

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

## POČÍTAČOVÁ PODPORA PRO MAPOVÁNÍ V TERÉNU

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Ondřej Hanák

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INTELLIGENT SYSTEMS

# POČÍTAČOVÁ PODPORA PRO MAPOVÁNÍ V TERÉNU

COMPUTER SUPPORT IN MAPPING SURVEYS

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Ondřej Hanák

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Martin Hrubý, Ph.D.

BRNO 2010

## **Abstrakt**

Práce se zabývá tvorbou aplikace pro kapesní počítače sloužící k podpoře mapování v terénu. Téma spadá do oblasti geografických informačních systémů (GIS), text proto začíná rozбором prerekvizitních témat jako jsou právě GIS, charakter zpracovávaných dat nebo principy globálního určování polohy (GPS). Pojednává o dostupných platformách kapesních počítačů z hlediska rozšíření a dostupnosti. Na základě rozboru dostupné komerční aplikace specifikuje požadavky na vyvíjený, volně dostupný, program. Následuje popis návrhu a některých implementačních detailů. Závěr je věnován ověření funkčnosti při mapování reálného geoobjektu.

## **Klíčová slova**

GIS, GPS, NMEA, geodata, prostorová data, mapování, PDA, smartphone, Windows Mobile

## **Abstract**

The report deals with creating applications for Pocket PC, designed to aid with terrain mapping process. Topic falls within the field of geographic information systems (GIS), so the text begins with an analysis of related topics such as GIS itself, processed data characteristics or global positioning systems (GPS). Available handheld platforms are discussed and based on analysis of existing commercial application, the requirements on developed program are set. Description of application architecture and some implementation details follow. The conclusion is devoted to testing functionality by mapping the real geospatial object.

## **Keywords**

GIS, GPS, NMEA, geodata, spatial data, mapping, PDA, smartphone, Windows Mobile

## **Citace**

Hanák Ondřej: Počítačová podpora pro mapování v terénu. Brno, 2010, diplomová práce, FIT VUT v Brně.

# Počítačová podpora pro mapování v terénu

## Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením

Ing. Martina Hrubého, Ph.D.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....  
Ondřej Hanák  
24. 5. 2010

© Ondřej Hanák, 2010.

*Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.*

# Obsah

Obsah.....	2
1 Úvod.....	4
2 Vymezení oblastí zájmu.....	6
2.1 GIS.....	6
2.1.1 Geoobjekty a geodata.....	6
2.1.2 Tematické vrstvy .....	7
2.1.3 Složky GIS.....	10
2.2 Získávání geodat.....	10
2.2.1 Geodetické měření v terénu .....	11
2.2.2 GPS .....	12
2.2.3 NMEA.....	14
2.3 Zpracování geodat .....	15
2.3.1 GRASS .....	15
2.4 Digitální zápisník.....	17
2.4.1 Kapesní počítač.....	17
2.4.2 Dostupné platformy .....	18
2.4.3 Windows Mobile .....	20
3 Existující řešení.....	23
3.1 ArcPad .....	23
3.2 Opodstatněnost další aplikace.....	25
4 Návrh aplikace .....	26
4.1 Upřesnění požadavků.....	26
4.2 Vývojové prostředky .....	27
4.3 Souborové formáty .....	29
4.3.1 Rastry.....	29
4.3.2 Vektory .....	30
4.3.3 Projektové a konfigurační soubory .....	32
4.4 Uživatelské rozhraní .....	33
5 Implementace.....	37
5.1 Architektura .....	37
5.1.1 Přehled tříd.....	38
5.2 Datové struktury .....	40
5.2.1 Geometrie .....	40
5.2.2 Atributy.....	42

5.3	Vybrané algoritmy .....	43
5.3.1	Výpočet vzdálenosti bodu od úsečky .....	43
5.3.2	Test polohy bodu uvnitř polygonu .....	44
5.3.3	Určení orientace posloupnosti bodů .....	45
5.4	Metriky kódu .....	45
6	Praktické testování .....	46
	Závěr .....	50
	Literatura .....	51
	Seznam příloh .....	53
	Příloha 1: kompletní diagram tříd .....	54
	Příloha 2: snímky obrazovek aplikace .....	56

# 1 Úvod

Cílem této práce je vytvořit softwarovou aplikaci pro kapesní počítače sloužící pro podporu geografického mapování v terénu. Řešené téma spadá do oblasti GIS – geografických informačních systémů. I přesto, že geografické informační systémy coby informatický obor představují relativně mladou, moderní a stále se rozvíjející oblast, stihly se už v mnoha oblastech lidské činnosti prosadit. Klasickým sektorem může být územní plánování, tvorba koncepcí rozvoje měst i zemědělských oblastí, řízení energetických a vodohospodářských soustav či řešení krizových situací a podpora fungování složek integrovaného záchranného systému. Nově se uplatňuje i v oblastech na první pohled nesouvisejících, například v marketingu. Kde postavit nové obchodní centrum? Kam umístit billboard, aby kolem projelo co nejvíce lidí? I na takové otázky mohou GIS pomoci odpovědět.

Náplň práce nicméně spadá do klasické podoblasti, kterou je mapování. Tato činnost sama o sobě, na rozdíl od GIS, je stará zcela nepochybně. Tvorbou map a způsoby, jak se v prostoru orientovat se lidé zabývali odjakživa. Spojení s moderními technologiemi tento obor jen posunulo dále. Běžně dnes můžeme pracovat s leteckými a družicovými snímky Země nebo využívat pohodlí globálních systémů pro určování polohy a pro navigaci. I přes dostupnost všech těchto technických vymožeností je někdy potřeba vyrazit ven a určitá data získat přímo na místě pomocí klasických prostředků „tužka a papír“. Pomoci právě v takovýchto případech při mapování v terénu má za úkol dále probíraná aplikace.

Druhá kapitola se zabývá úvodem do oblasti geografických informačních systémů a související problematikou. Definuje základní pojmy a uvádí potřebné souvislosti. Přes geoobjekty a geodata se dostává k jejich sdružování do tematických vrstev. U obou typů – rastrových i vektorových – demonstruje různé způsoby reprezentace a uložení dat. Jako jednu z možností sběru geodat přibližuje družicový navigační systém GPS a komunikační protokol NMEA, pomocí kterého jsme schopni geografické údaje z přijímače přečíst a dále zpracovávat. Nastiňuje rozdílné požadavky na nástroje pro sběr a pro následné zpracování údajů. Zavádí pojem digitálního zápisníku ve formě softwarové aplikace pro kapesní počítač. U několika systémů kapesních přístrojů rozebírá jejich klady a zápory z hlediska technické výbavy, rozšířenosti mezi uživateli, cenové dostupnosti a programátorské podpory. Končí výběrem vhodné platformy a představením referenčního přístroje, na kterém bude vývoj probíhat.

Idea využití mobilní výpočetní techniky coby digitálního geodetického zápisníku není sama o sobě průlomová. Pro zvolenou platformu Windows Mobile jsou dostupné komerční programy sloužící uvedenému účelu. Vlastnostmi a představením kladů a záporů patrně nejpropracovanějšího programu ArcPad se zabývá třetí kapitola. Diskutuje opodstatnění vývoje vlastního softwarového produktu a ukazuje výhody, které nová aplikace uživatelům přinese.

Čtvrtá kapitola se věnuje celkovému návrhu programu. Nejdříve sumarizuje požadavky vzešlé z předchozích dvou kapitol a dále se zabývá volbou vývojových prostředků. Vybrán byl jazyk C# ve spolupráci s Compact .NET Frameworkem, jako integrované vývojové prostředí Microsoft Visual Studio 2008, které mimo jiné poskytuje emulátor mobilního zařízení. Protože aplikace nebude uzavřená, ale bude komunikovat s jinými programy a systémy, je další podkapitola věnována výběru souborových formátů k ukládání a načítání dat. Pro vektorová data to je standardizovaný shapefile, pro fotomapy několik typů běžných rastrových obrazových formátů. Nakonec je popsána struktura a základní funkce uživatelského rozhraní – rozložení ovládacích prvků jednotlivých obrazovek včetně jejich návazností.

Kapitola pátá se zabývá konkrétní implementací v jazyce C# založené na návrhu z kapitoly čtvrté. Začíná celkovým pohledem na architekturu, rozdělení na jednotlivé moduly a organizaci tříd. Probírá zásadní datové struktury pro reprezentaci geometrie (bod, linie a polygon) a atributů (datový sloupec a řádek). Podrobněji představuje několik vybraných algoritmů pro určování vzdálenosti objektů nebo orientace vrcholů polygonu. Kapitulu uzavírá základní přehled metrik kódu a výsledných binárních souborů.

Nedílnou součástí vývoje prakticky zaměřeného programu je i jeho testování v reálných podmínkách. Účelem je ověřit, zda navržené uživatelské rozhraní programu umožňuje efektivní ovládání všech funkcí a zda je výkon a odezva aplikace dostatečná. To je náplň šesté závěrečné kapitoly.



## 2 Vymezení oblastí zájmu

Tato kapitola se zabývá úvodem do oblasti geografických informačních systémů a související problematikou. Definuje základní pojmy a uvádí potřebné souvislosti. Přes geoobjekty a geodata se dostává k jejich sdružování do tematických vrstev. U obou typů – rastrových i vektorových – demonstruje různé způsoby reprezentace a uložení dat. Jako jednu z možností sběru geodat přibližuje družicový navigační systém GPS a komunikační protokol NMEA, pomocí kterého jsme schopni geografické údaje z přijímače přečíst a dále zpracovávat. Nastiňuje rozdílné požadavky na nástroje pro sběr a pro následné zpracování údajů. Zavádí pojem digitálního zápisníku ve formě softwarové aplikace pro kapesní počítač, jehož vývoj je cílem této práce. U několika systémů kapesních přístrojů rozebírá jejich klady a zápory z hlediska technické výbavy, rozšířenosti mezi uživateli, cenové dostupnosti a programátorské podpory. Končí výběrem vhodné platformy a představením referenčního přístroje, na kterém bude vývoj probíhat.

### 2.1 GIS

S mapováním za podpory výpočetní techniky se v dnešní době úzce pojí zkratka GIS neboli geografické informační systémy. Je tedy důležité si na začátku vymežit, co tento pojem zahrnuje. Jednotná a ustálená definice zatím neexistuje, autoři v různých pramenech uvádějí více či méně odlišné varianty. Jako nejkomplexnější se jeví definice z [1]:

*GIS je na počítačích založený informační systém pro získávání, obhospodařování, analýzu, modelování a vizualizaci geoinformací. Geodata, která využívá, popisují geometrii, topologii, tematiku (atributy) a dynamiku (změny v čase) geoobjektů.*

#### 2.1.1 Geoobjekty a geodata

Geografickou informací chápeme údaj o hmotném nebo nehmotném objektu, kde její primární součástí je údaj o geografické poloze objektu. Geodata jsou veškerá data, se kterými GIS pracují a skládají z jednotlivých geoobjektů. Geoobjekt je část zkoumaného prostoru, kterou je možno na zvolené úrovni generalizace modelovat jako jeden objekt (například celé město můžeme v určité situaci abstrahovat na jeden jediný bod). Geoobjekt může obsahovat dva základní druhy informací:

- **prostorové** (anglicky *spatial*) informace – tvar, poloha, topologie
- **neprostorové** informace – veškeré další relevantní atributy, které zkoumaný objekt charakterizují jinak než polohově

Abychom se mohli na geoobjekt odvolávat, obvykle mu přidáváme nějakou identifikaci (typicky název – Brno, Svatka) a v některých případech i časový údaj, umožňující nám sledovat změny v čase.

Například pro hmotný geoobjekt rozhledna můžeme zpracovávat její polohu (coby prostorový údaj) a navíc rozšiřující atributy jako stáří, použitý stavební materiál nebo třeba jméno autora architektonického návrhu.

Prostorové údaje můžeme dále rozdělit na geometrii a topologii.

**Geometrie** je prostorový popis objektu. Zahrnuje jeho polohu, tvar a velikost. Geometrickým vztahem dvou objektů je například jejich vzdálenost.

**Topologie** se nezabývá prostorovým popisem, ale výhradně logickými vazbami mezi objekty. Lze ji vnímat jednak při zkoumání vazeb mezi objekty na úrovni atributu, jednak při organizování uložení geoobjektu do databáze, kde cílem je rychlé provádění analytických algoritmů v mapových vrstvách. Právě toto hledisko je zásadní při tvorbě GIS aplikací. S tímto pohledem se také pojí dělení dat podle počtu dimenzí – počtu rozšíření objektu do prostoru podle souřadných os. Reálné objekty na Zemi jsou sice vždy trojrozměrné, do prostředí GIS se však transformují podle potřebné úrovně generalizace:

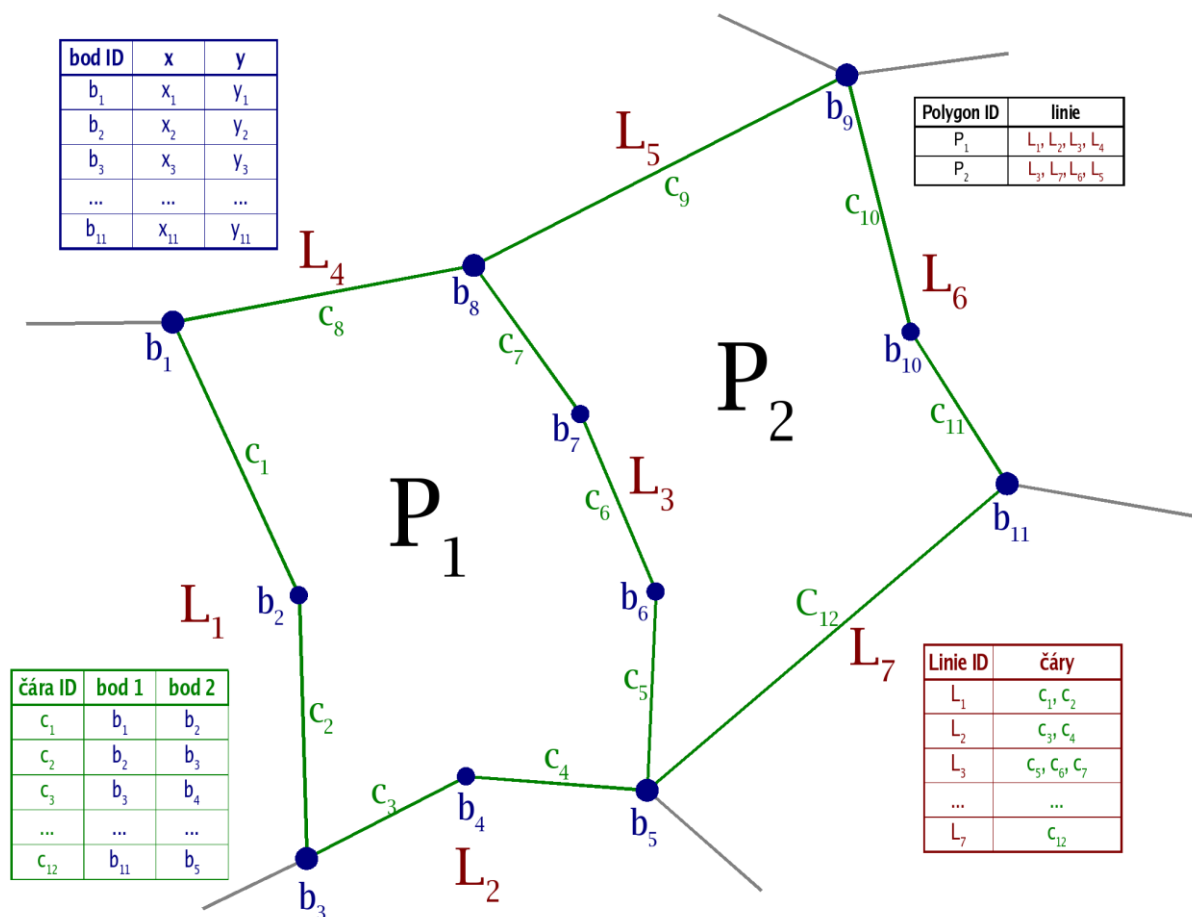
- **Bezrozměrné** 0D geoobjekty, body, definované pouze svou polohou. Příkladem může být už zmíněná rozhledna nebo třeba studna.
- **Jednorozměrné** 1D geoobjekty, úseky čar s konečnou délkou. Pomocí 1D geoobjektů se nejčastěji modelují například silnice, řeky a obecně jakékoliv sítě.
- **Dvojrzměrné** 2D geoobjekty, polygony, uzavřené rovinné útvary.
- **Trojrozměrné** 3D geoobjekty, polyhedrony. V GISech se používají výjimečně. Třetí rozměr je nejčastěji modelován pomocí tzv. *digitálního modelu terénu* (anglicky *digital elevation model*, DEM).

## 2.1.2 Tematické vrstvy

Pro praktické zacházení je vhodné geodata třídit do tematických souborů, v programech obvykle nazývaných mapové hladiny či vrstvy, nebo přesněji tematické mapové vrstvy – například vodstvo, silnice nebo rozložení půdních typů. Smyslem takové dělení je usnadnění analýzy dat a selektivní práce s nimi. Právě analýza dat je snad nejčastějším důvodem k nasazení GISu pro modelování reality. Vrstvy se podle povahy a způsobu uložení modelovaných dat dělí na dva typy – vektorové a rastrové.

Ve **vektorových vrstvách** jsou data uložena pomocí bodů a čar. Bod je základním bezrozměrným elementem s definovanou polohou. Čára je úsečka nebo křivka spojující dva body.

Z důvodů zjednodušení se křivky běžně reprezentují jako série úseček. Při modelování geodat pomocí vektorů se hojně využívá teorie grafů. Vektorová data mohou být v GISech organizována a ukládána různými způsoby.



Obrázek 1: demonstrace uložení vektorových dat za použití hierarchického modelu.

prevzato z [http://cs.wikipedia.org/wiki/Soubor:GIS\\_hierarchic\\_model.png](http://cs.wikipedia.org/wiki/Soubor:GIS_hierarchic_model.png)

Ve **špagetovém modelu** jsou všechny typy objektů (bez ohledu na počet dimenzí) uloženy v jednom heterogenním seznamu, jehož prvky mají pouze 2 položky:

- typ objektu – bod, čára, polygon
- parametry objektu – jedna či více souřadnic

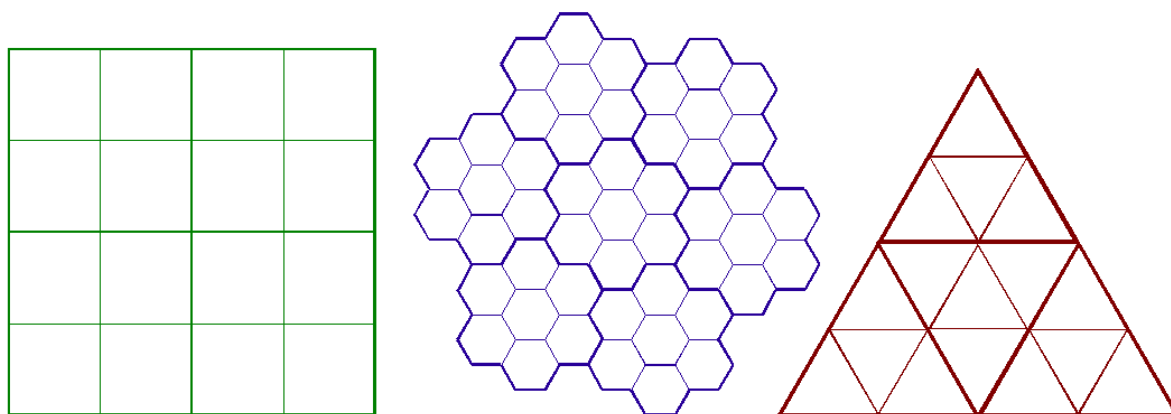
Špagetový model neobsahuje informace o topologii (sousednost, orientace, konektivita, obsahování) a proto je tento model pro analýzu geodat obtížně použitelný. Navíc zde dochází k redundanci dat (překrývání bodů). Výhodou je snadná manipulace s daty na programové úrovni.

**Hierarchický model** data ukládá hierarchicky s ohledem na počet dimenzí. Vychází z faktu, že polygon se skládá z několika linií, linie z několika úseček a úsečky jsou spojením dvou bodů. Tyto elementy jsou pak uloženy samostatně. Princip je znázorněn na obrázku Obrázek 1.

**Topologický model** je kompromis mezi špagetovým a hierarchickým modelem. Ukládají se pouze body a čáry, přičemž k čáře lze připojit informaci o její orientovanosti, z čehož pak lze odvodit sousedící polygony.

**Rastrové vrstvy** se používají k modelování veličin, které jsou spojitě definovány na celém zkoumaném prostoru. Například vrstva nadmořské výšky, rozložení půdních typů, atmosférického tlaku a podobně.

Prostor je v tomto případě rozdělen na menší části, jejichž rozměr je dostatečně malý na to, aby na nich bylo možné hodnotu dané veličiny považovat za konstantní. Dělení prostoru může být buď pravidelné, nebo nepravidelné, buňky mohou být různého tvaru (čtverec, trojúhelník, šestiúhelník), viz Obrázek 2. V naprosté většině případů se ale používá pravidelné dělení čtvercovou mřížkou.



Obrázek 2: různé typy rastrů.

převzato z [http://cs.wikipedia.org/wiki/Soubor:Type\\_of\\_rasters.png](http://cs.wikipedia.org/wiki/Soubor:Type_of_rasters.png)

Speciálním případem rastrové vrstvy jsou fotomapy. Jedná se o snímek zemského povrchu, který je takzvaně georeferencován – má jednoznačně určeno, jakou oblast zobrazuje. To umožňuje jeho přesné slícování s obsahem vektorových vrstev. Typické užití fotomap je v roli pomůcky pro zvýšení názornosti při vytváření a editaci dalších vrstev. Jako zdroj fotomap nejčastěji slouží naskenované papírové mapy a letecké či družicové snímky.

### 2.1.3 Složky GIS

Dále tedy budeme chápat GIS jako systém elektronický, založený na využití výpočetní techniky. Jako každý jiný informační systém zahrnuje i tento několik komponent:

- **Hardware** – technické prostředky, s jejichž pomocí získáváme, uchováváme a zpracováváme geodata. Pro získávání dat využíváme mapovací a geodetické pomůcky, pro zpracování a uchovávání pak více či méně standardní výpočetní techniku, typicky osobní počítač, skener pro možnost vstupu a následnou *digitalizaci analogových* dat, tiskárnu či plotter pro možnost mapového výstupu.
- **Software** – specializované programy pro zpracování a vizualizaci geodat.
- **Data** – nejdůležitější a často také finančně nejnáročnější součást GISu. Může se jednat o data získaná vlastním měřením či z nějaké již existující (tematické) databáze.
- **Uživatelé** – lidé se znalostmi geografie, kteří za pomoci HW a SW pracují s geodaty.

Všechny tyto komponenty jsou nerozlučně spjaty: o této diplomové práci je možné prohlásit „*úkolem je vyvinout softwarovou aplikaci, pomocí které budou uživatelé na mobilním hardwaru pracovat s daty*“.

K získávání dat se vážou pojmy analogový, digitální a digitalizovaný, které mají v technice široký význam, nicméně v tomto případě označují původ a způsob uložení geografických dat. Vycházejme z faktu, že fyzikální vlastnosti objektů okolo nás jsou spojité, možná poněkud netechnicky řečeno nekonečně přesné. Získáváním *analogových* údajů můžeme rozumět tradiční postup například zeměměřičů, kteří ve zkoumané lokalitě provádějí potřebná měření bez využití výpočetní techniky. Výsledky zapisují nejčastěji na papír a odpovídajícím způsobem s nimi dále pracují. *Digitální* údaje jsou naopak už od svého vzniku vázány na výpočetní techniku. Důležité je, že dalším zpracováváním neztrácejí na přesnosti a kvalitě. Takové údaje můžeme získávat i automaticky, bez zásahu člověka. *Digitalizovaná* data vzniknou pozdější digitalizací původně analogových údajů. Digitalizace tedy znamená převedení původní spojité informace na konečné číselné vyjádření. Tento postup je častou součástí životního cyklu geodat. Digitalizovaná data bývají nazývána jako *sekundární zdroj dat*, viz dále.

## 2.2 Získávání geodat

Ukazuje se, že nejhodnotnější součástí GISů jsou právě data. Naplňování databáze daty je v drtivé většině případů nejnáročnějším krokem v rámci GIS projektu jak časově, tak finančně. Podle [2] tvoří náklady na pořizování a údržbu dat až 80% celkového rozpočtu.

Při získávání dat je potřeba použít techniku a metodu odpovídající jak požadované přesnosti výstupu, tak finančním možnostem projektu. Způsoby získávání dat je možné rozdělit na primární a sekundární.

Mezi **primární zdroje** patří takové metody, kdy vznikají nová, původní data. Tento způsob je obvykle pracný a tedy i drahý. Bývá volen v případech, kdy potřebná data nemůžeme získat jinak (koupit), nebo kdy nejsou dostatečně důvěryhodná. Spadají sem veškerá geodetická měření v terénu (produkuje vektorová data, typicky pro mapy velkých měřítek), fotogrametrické metody (zpracování fotografických snímků produkující rastrová data) a vstup z DPZ (dálkový průzkum Země, zpracování informací z leteckých a družicových nosičů, produkuje rastrová data typicky ve velkých měřítcích). Poslední, široce dostupnou a levnou možností produkující vektorová data je vstup údajů z *GPS* (*Global Positioning System* – globální polohovací systém). Podrobnější pojednání o této technologii následuje dále v podkapitole 2.2.2.

**Sekundární zdroje** jsou již jednou zpracované primární zdroje. Nejčastěji se taková data získávají digitalizací analogové předlohy nebo nákupem. Na shromažďování a následný prodej se zaměřují specializované instituce. Nevýhodou sekundárních zdrojů může být případná menší přesnost (způsobená procesem digitalizace) nebo nedostupnost přesně takových dat, jaká požadujeme. Zde pak hraje roli, zda jsme ochotni odchylky tolerovat, nebo zda se nám už vyplatí získat vlastními silami data primární.

## 2.2.1 Geodetické měření v terénu

Geodetické měření je v současné době pravděpodobně nejpresnější a nejspolehlivější způsob získání (geometrické) informace o zkoumané lokalitě. Dle rozdělení výše se jedná o primární metodu získávání geodat a to přímo na místě zájmu. Při takovém měření obvykle využíváme nějakou formu geodetického zápisníku (dále jen zápisník), do kterého zaznamenáváme získané údaje, což může být:

1. Geografická poloha zaměřeného bodu – například pomocí prostředků globálního zjišťování polohy (GPS, viz dále). Měřicí přístroj může být přímo součástí zápisníku.
2. Hodnoty zkoumaných atributů v zaměřeném bodě.
3. Volitelně čas a další okolnosti měření, jako vnější vlivy (potenciálně) ovlivňující přesnost měření.

Dnes často jako elektronický zápisník slouží kapesní přístroje typu PDA či smartphone. Protože sběr dat probíhá v terénu, někdy v místech se ztíženým přístupem nebo za zhoršených povětrnostních podmínek, musí být práce se zápisníkem co možná nejjednodušší. Předpokládá se také, že pokročilejší manipulace se získanými daty jako například analýzy nebo tvorba mapového díla probíhají až po návratu z měření s využitím pokročilejších technických prostředků (PC s velkým

monitorem a plnohodnotnými vstupními zařízeními). Proto i funkcionality zápisníku bývá oproti desktopovým systémům omezena.

## 2.2.2 GPS

*Global Positioning System*, zkráceně GPS, je původně vojenský navigační systém provozovaný Ministerstvem obrany Spojených států amerických. Plný název je ve skutečnosti *Navigation Signal Timing and Ranging Global Positioning System*, tedy NAVSTAR GPS, avšak běžně se pod krátkým (a veřejně rozšířeným) označením GPS myslí právě NAVSTAR GPS.

GPS je tedy prostředek globální navigace, pomocí kterého lze s několikametrovou přesností určit pozici uživatele kdekoli na Zemi (na či nad povrchem). Přesnost lokalizace lze určitými způsoby (např. s využitím metody *Differential GPS*) zvýšit až na jednotky centimetrů. Vývoj systému probíhal v letech 1973 až 1994, kdy byla ve vesmíru umístěna kompletní sestava satelitů. Ze strategických důvodů byla do roku 2000 do civilního signálu vnášena umělá chyba (označení *SA - Selective Availability*, česky též selektivní dostupnost).

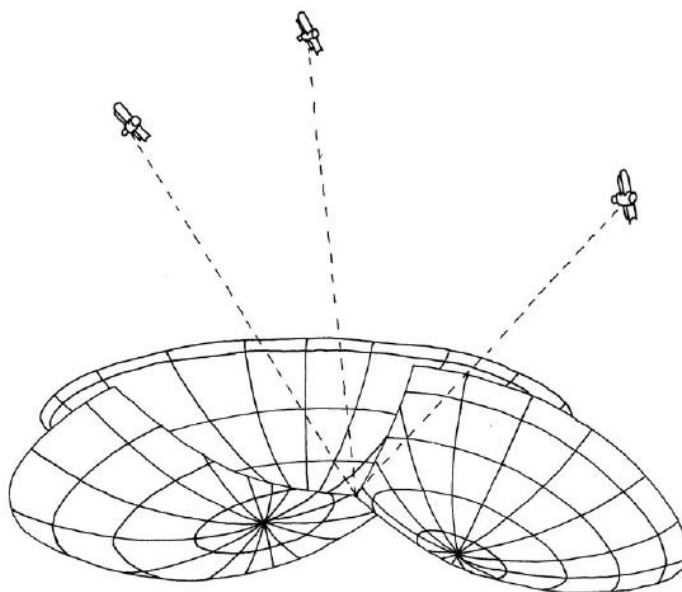
V současné době se systém využívá i v mnoha oborech lidské činnosti v civilním sektoru. I když na podle [3] se na provoz GPS ročně vynakládá 600 až 900 milionů amerických dolarů, mohou ho všichni zájemci využívat zdarma.

Vesmírnou část systému tvoří družice obíhající ve výšce okolo 20 300 km nad Zemí na šesti oběžných drahách. Na jedné dráze se pohybuje čtyři až šest družic. V jednom okamžiku je současně aktivních 24 družic, zbytek do 32 tvoří zálohu. Díky pohybu družic se jejich konfigurace na obloze neustále mění, z každého místa na Zemi vždy bývá přímo viditelných 6 až 12 družic. Jeden oběh jim trvá 12 hodin.

Z uživatelského hlediska pracuje GPS pouze jednosměrně, tedy aktivní družice vysílají a pasivní pozemní stanice přijímají. Pro přenos signálů z družic je vyhrazeno několik kanálů na frekvencích zhruba 1175 až 1675 MHz. Díky velké vzdálenosti vysílačů od zemského povrchu a jejich omezeným zdrojům energie ze solárních panelů je síla přijímaného signálu velice malá, udává se řádově  $10^{-16}$  wattů. Takto slabý signál podléhá atmosférickým vlivům a je obtížného ho rozlišit od všudypřítomného elektromagnetického šumu. Proto kapesní přijímače fungují prakticky jen s přímým výhledem na oblohu, uvnitř budov bývají nepoužitelné. Pro zlepšení kvality příjmu je možné u některých modelů připojit externí anténu.

Uživatelská část je tvořena přijímači signálu z družic, které jsou v danou chvíli nad obzorem. Na základě přijatých dat přijímač vypočítá svou polohu, nadmořskou výšku a případně i chybovou odchylku těchto údajů. Nejdůležitější součástí dat je zpráva o poloze vysílače daná tzv. *efemeridou*, což je astronomické vyjádření polohy vesmírného tělesa, dále obsahuje přesný údaj o čase a odhad zpoždění signálu v ionosféře. Další složkou je tzv. *almanach*, což je přibližná informace o orbitech

dalších satelitních stanic. Z těchto údajů přijímač rychleji odvodí, jaké satelity se právě nachází v jeho zorném poli a rychleji se „zorientuje“.



Obrázek 3: grafické znázornění principu trilaterace podle [7].

Samotné zjištění polohy z těchto informací se nazývá *trilaterace*. Je to metoda určování polohy objektů pomocí vztahů v trojúhelníku podobně jako u triangulace. Na rozdíl od ní se však při výpočtu polohy nevyužívá změřených úhlů ke známým referenčním bodům, ale vzdálenosti od nich. K jednoznačnému určení polohy bodu pomocí trilaterace jsou potřeba alespoň tři referenční body. Čím více bodů známe, tím přesnější je výpočet. Referenčními body jsou samotné družice, jejichž poloha je obsažena ve vysílaných datech. Vzdálenost se poté určí z doby šíření signálu od družice k přijímači. U rychlosti šíření signálu se počítá s rychlostí světla a případnou korekcí založenou na aktuálním stavu ionosféry (opět součást vysílaných dat). Jakmile zná přijímač vzdálenost k jedné družici, lokalizuje svou polohu někde na plášti koule okolo této družice s poloměrem rovným vzdálenosti od družice. Aplikování stejného postupu na druhou družici zpřesní svou polohu na kružnici tvořenou průnikem dvou pomyslných koulí. Třetí koule omezí možnou pozici pouze na dva body, z nich jeden obvykle leží vysoko nad povrchem Země a je tedy vyloučen jako nepravděpodobný – viz Obrázek 3: grafické znázornění principu trilaterace podle [7]. Čím více údajů z různých satelitů je do výpočtu zahrnuto, tím přesněji je poloha určena.

Vzhledem k uvedeným vlastnostem jako je dosažitelná přesnost a cena budeme dále s GPS počítat jako s hlavním zdrojem polohové složky pro zpracovávání geodata.



### 2.2.3 NMEA

*NMEA 0183* nebo zkráceně jen NMEA je standard popisující jak elektronickou, tak i datovou formu komunikace mezi GPS přijímačem (anglicky *talker*) a klientským zařízením zpracovávajícím data (anglicky *listener*).

Protokol byl vyvinut americkou firmou *National Marine Electronics Association* a jedná se o komerční produkt, specifikace se na webu oficiálním webu organizace, [www.nmea.org](http://www.nmea.org), prodává v elektronické podobě za 300 USD (ke 2. 12. 2009). Pravděpodobně díky reverznímu inženýrství byl formát komunikace odhalen a jeho popis se dá nalézt na internetu. Následující informace nevychází z oficiální specifikace, ale ze zdrojů [4] a [5] volně dostupných na webu. V praxi se ukázalo, data získaná uvedenými postupy odpovídají realitě, proto budeme uvedené zdroje pro potřeby této práce pokládat za dostatečně důvěryhodné.

Elektronická část přenosu je specifikována jako sériová dle standardu RS-232 s následujícím nastavením:

- přenosová rychlost 4800 baudů [Bd]
- počet datových bitů 8
- počet stop bitů 1 nebo 2
- parita žádná

Co se týče přenosové rychlosti, výrobci GPS přijímačů obvykle nabízejí i vyšší hodnoty, nicméně uvedených 4800 baudů představuje univerzální a zaručenou hodnotu.

Z hlediska datového neboli obsahového je komunikace tvořena jakýmsi balíky či bloky dat označovanými jako věty, anglicky *sentences*. Věty jsou tvořeny posloupností ASCII znaků, kde každá začíná znakem dolaru \$, pokračuje pěti písmeny, z nichž první dvě označují druh zařízení (GP pro GPS) a zbylá tři typ věty. Dále následují datová pole oddělená čárkami. Význam těchto datových polí se liší podle typu věty. Pokud není hodnota pro některé pole aktuálně dostupná, zůstává prázdné, což se projeví jako dvě čárky následující bezprostředně po sobě. Za posledním polem může bez čárky navazovat znak hvězdička \* a dvoumístné hexadecimální číslo jako kontrolní součet, konkrétně XOR všech hodnot bajtů mezi \$ a \*. Tento kontrolní součet je pro některé typy vět povinný, většinou ale volitelný. Věta je ukončena koncem řádku – znaky <CR> <LF>, hexadecimálně 0xD a 0xA. Maximální délka věty včetně \$ a konce řádku činí 83 znaků. Věty jsou vysílány jednou za sekundu.

Jedna konkrétní věta může vypadat třeba takto:

```
$GPGSA,A,3,04,05,,09,12,,,24,,,,,2.5,1.3,2.1*39
```

Typů vět existuje velké množství, v praxi se ale nejčastěji používají následující 4:

1. **GSA** (*Active Satellites*) – informace o aktuálně využívaných satelitech.
2. **RMC** (*Recommended Minimum Navigation Information*) – minimální doporučená informace pro navigaci, určení zeměpisné polohy a času.
3. **GSV** (*Satellites in View*) – informace o viditelných satelitech.
4. **GGA** – zeměpisná délka a šířka, nadmořská výška, čas určení souřadnic.

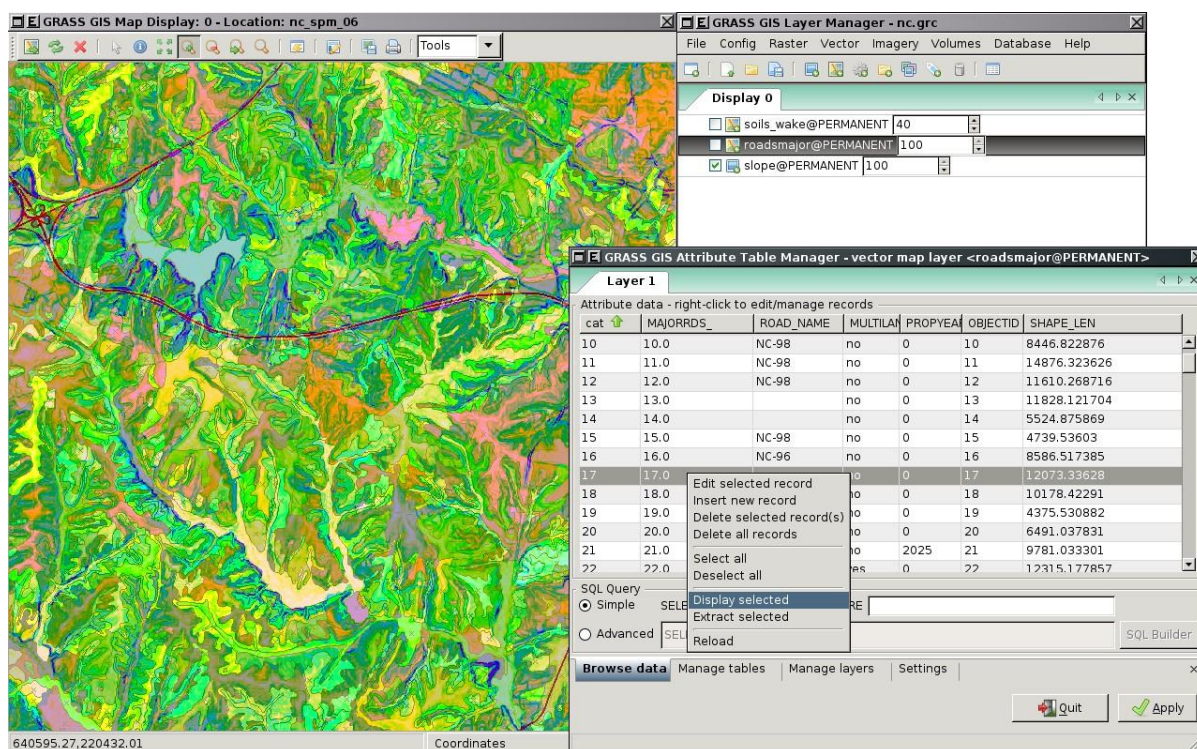
## 2.3 Zpracování geodat

Možnosti získání dat jsme již probrali, další fází je tedy jejich zpracování. Budeme tím myslet jejich ukládání, transformace a analýzy na PC s pomocí vhodného desktopového programu. Takových softwarových produktů existuje na trhu nepřeberné množství, od jednoúčelových utilit až po komplexní produktové balíky. Ty jsou dostupné jednak jako komerční produkty, jednak coby freeware nebo dokonce opensource, tedy včetně kompletních zdrojových kódů. Výrazným zástupcem komerční rodiny je například balík ArcGIS severoamerické společnosti ESRI (<http://www.esri.com>), jehož cena odpovídá komplexnosti – řádově se jedná o tisíce amerických dolarů. Zástupcem druhé, bezplatné, kategorie je český produkt Janitor firmy Cenia (<http://janitor.cenia.cz>) nebo celosvětově rozšířený opensource systém GRASS (<http://grass.itc.it>).

### 2.3.1 GRASS

*Geographic Resources Analysis Support System*, zkráceně *GRASS* je otevřený geografický informační systém šířený pod licencí GNU GPL. Lze ho provozovat na mnoha platformách – MS Windows, GNU/Linux i Mac OS. Umožňuje pracovat s rastrovými i vektorovými geografickými daty a to jak pomocí příkazové řádky, tak s využitím grafického uživatelského rozhraní.

Vývoj produktu byl původně započat ve výzkumných laboratořích (*Construction Engineering Research Lab*, zkráceně *CERL*) americké armády v roce 1982. I přes to, že projekt stál řádově miliony amerických dolarů, byl koncem 80. let minulého století poskytnut veřejnosti, a to včetně zdrojových kódů. CERL se na vývoji podílel do roku 1995, v roce 1997 pak vznikl *GRASS Development Team*, který se soustředil na Baylor University v Texasu v USA a na Universitát Hannover v Německu. Z armádního projektu se tedy GRASS rozšířil nejdříve do akademické sféry a dnes už nachází uplatnění i ve sféře komerční – mezi významné uživatele GRASSu patří například americké instituce NASA (*National Aeronautics and Space Administration*, česky Národní úřad pro letectví a kosmonautiku) nebo NOAA (*National Oceanic and Atmospheric Administration*, česky Národní úřad pro oceán a atmosféru).



Obrázek 4: uživatelské rozhraní systému GRASS.  
převzato z <http://grass.osgeo.org/screenshots/gui.php>

Český překlad původní německé příručky z roku 2003 „GIS GRASS – Praktická rukověť“ [6] hovoří o GRASSu takto:

GRASS je kombinovaný rastrový a vektorový GIS s integrovaným systémem pro správu obrazových dat a vizualizaci. Obsahuje přes 400 programů a pomocných prostředků, sloužících k práci s rastrovými, vektorovými a bodovými daty, vytváření map v digitální a analogové formě, vytváření a ukládání prostorových dat. Vedle graficky orientovaného uživatelského rozhraní disponuje GRASS i textovou příkazovou konzolí. GRASS lze připojit na tiskárnu, plotter a digitalizační prkno. Jeho prostřednictvím je možné přistupovat k datům z externích databází. Je proto ideální pro plánování krajiny a inženýrsko-technické použití. Stejně jako jiné GISové balíky, umožňuje i GRASS práci s vektorovými daty reprezentujícími silnice, vodní toky, hranice a jiné objekty. Prostřednictvím integrovaných digitalizačních funkcí lze aktualizovat vektorové mapy. GRASSové moduly umí provést konverzi mezi vektorovým a rastrovým formátem. GRASS dále obsahuje nástroje, které z něj vytváří plnohodnotný on-line GIS, přístupný přes World-Wide-Web.

Moduly pro práci s obrazovými daty jsou srovnatelné se špičkovými produkty tohoto sektoru. Mnohdy jsou i bohatší než proprietární GISové balíky. Obsahují množství nástrojů sloužících ke zpracování a vyhodnocování multispektrálních satelitních dat, stejně jako moduly pro produkci ortogonálních map z naskenovaných leteckých snímků. GRASS tak umožňuje využít takřka veškerých možných cest, vedoucích k importu dat do GISu.

Kromě standardní dvojrozměrné analýzy dovoluje GRASS zpracovávat data i ve třech dimenzích. Rastrová, vektorová a bodová data lze použít při vizualizaci. Příkladem tohoto použití může být návrh letiště, analýza krajiny a prostorových náchylností. Vizualní prostředky umožňují animaci prostorových dat. 3D pohledy tak mohou být prezentovány jako jednotlivé obrázky nebo jako MPEG film a uloženy pro další práci. GRASS obsahuje rovněž soubor modelů z oblasti hydrologických analýz. Jedná se mimo jiné o vymezení povodí, počítání SCS křivek, analýzy povodňových vln a využití různých modulů pro kompletní simulaci povrchového odtoku z daného území. Další moduly mohou vytvářet diagramy a statistiky k modelovaným a kalibrovaným datům. Dále je možné v GRASSu pracovat s krajinnými daty a odvodit některé parametry na základě numerických dat.

Tolik výňatek z příručky. Je vidět, že možnosti aplikace jsou opravdu rozsáhlé. Navíc tento popis pochází z roku 2003, od té doby vývojový tým urazil další kus cesty a dnešní možnosti jsou ještě bohatší. Možnost spolupráce s tímto otevřeným systémem proto bude zahrnuta mezi požadavky na výsledný program.

## 2.4 Digitální zápisník

Jak jsme již shrnuli v kapitole 2.2, počáteční fází životního cyklu geodat je jejich pořízení. Jednou z primárních možností je v tomto případě jejich získávání přímo v terénu za pomoci geodetického zápisníku. Protože GIS jsme si pro potřeby této práce vymezili jako systém elektronický, požadujeme i elektronický – digitální zápisník. Ten se skládá ze dvou částí – softwarová aplikace a hardware, na kterém běží. Požadavky na samotnou aplikaci rozeberu v kapitole 4, nyní se zaměříme na výběr fyzického zařízení.

### 2.4.1 Kapesní počítač

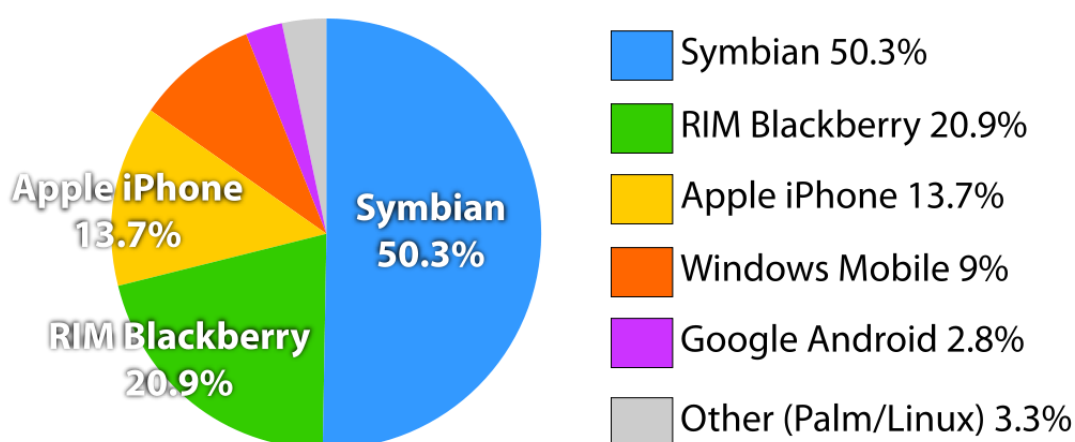
Základní hardwarovou částí digitálního zápisníku tvoří kapesní počítač. I když se počítače třídy notebook, netbook a výhledově i *tablet* neustále zmenšují, jejich rozměry, hmotnost, nízká fyzická odolnost a obvykle i nedostatečná výdrž provozu na baterie je z používání venku v přírodě diskvalifikují. Kapesním počítačem je proto myšleno zařízení typu *PDA* (*Personal Digital Assistant*). V některých pramenech se přístroje tohoto typu označují také jako palmtopy nebo handheldy (z anglického *palm* – dlaň a *hand* – ruka, čili volně přeloženo „do ruky“).

V posledních letech také probíhá, či snad už proběhla, úspěšná integrace PDA s mobilními telefony, takto vzniklá zařízení bývají označována jako komunikátory či smartphony („chytré telefony“). Klasická, „ne-telefonující“ PDA coby nové kusy jsou na trhu dostupná řádově jen v jednotkách modelů. Nicméně na internetových aukcích a v bazarech (a tedy i mezi lidmi) je k mání nepřeberné množství starších kusů. Důležitý pro nás je fakt, že dnešní smartphony funkcionalitu PDA rozšiřují, nikoliv omezují. V tomto ohledu proto můžeme mluvit o zpětné kompatibilitě.

Pod označením kapesní počítač budeme dále chápat oba typy zařízení, tedy jak klasické PDA bez telefonního modulu, tak smartphony.

## 2.4.2 Dostupné platformy

Trh s kapesními počítači je poměrně rozsáhlý a existuje tu silná konkurence jak mezi výrobci samotného hardware, tak operačních systémů, které na těchto zařízeních běží. Rozdělení trhu mezi operační systémy pro smartphony za druhý kvartál 2009 ukazuje Obrázek 5. Pokud bychom zkoumali jen operační systémy pro PDA, měl by Microsoft se svými Windows Mobile téměř 100% podíl, protože v tomto segmentu již nemá konkurenci. Jeho dřívější největší rival, společnost Palm Inc. se systémem Palm OS, tento segment v roce 2005 opustil a nadále se věnuje pouze vývoji smartphonů.



Obrázek 5: tržní podíl jednotlivých operačních systémů pro smartphony za druhý kvartál 2009.

převzato z [http://en.wikipedia.org/wiki/Mobile\\_operating\\_system](http://en.wikipedia.org/wiki/Mobile_operating_system)

Pro jakou platformu tedy zápisník navrhnout? Takovou, která splňuje několik základních požadavků:

1. Funguje na přístrojích s dotykovým displejem. Toto je předpoklad pro rychlý a intuitivní vstup dat a ovládání programu.
2. Je možné zajistit příjem dat o geografické pozici z GPS přijímače. Přístroj tedy musí mít buď GPS modul integrovaný uvnitř, nebo podporovat technologii Bluetooth pro připojení externího přijímače.
3. Na vybrané platformě musí existovat podpora tvorby aplikací v některém vyšším programovacím jazyce typu C# nebo Java. Musí existovat bezproblémový a bezplatný způsob distribuce hotových aplikací mezi uživatele.
4. Cílový systém musí být v rozumné míře rozšířený a musí být aktuálně dostupná nová zařízení, s reálnou vyhlídkou na existenci i v budoucnosti

V souladu s těmito body jsem z výběru v první řadě eliminoval systémy s nejnižším rozšířením, tedy stagnující Palm, experimentální mobilní linuxové distribuce (například systém *OpenMoko* na zařízení *FreeRunner*) a relativní novinku, systém *Google Android* (od roku 2007). Android je sice také postaven na linuxovém jádře a vyvíjen pod otevřenou GPL licenci, nicméně za jeho vývojem a koordinací stojí sdružení firem *Open Handset Alliance*, jehož členem je i společnost Google, která se na projektu významně mediálně i finančně podílí. Systém brzy získal i důležitý předpoklad pro rozšiřování, a to podporu různých výrobců hardware, kteří pro něj smartphony vyrábí. Nicméně systém je stále velmi mladý a zatím se dostatečně rozšířit nestihl, i když je mu předpovídána slibná budoucnost.

Zabývat se dále nebudeme ani operačním systémem *iPhone OS* na multimediálních přístrojích iPhone (smartphone) a iPod Touch (multimediální PDA). Jedná se o uzavřenou technologii společnosti Apple vycházející z jejich desktopového operačního systému MAC OS X. Firma vyvíjí jak vlastní hardware, tak operační systém, což na jedné straně přináší více možností optimalizace pro výkon, na druhé brzdí vývoj platformy díky neexistenci konkurence. Současně je pravda, že tento přístroj započal revoluci v chápání uživatelského rozhraní chytrých telefonů odstraněním prakticky všech hardwarových kláves a spolehnutím se téměř výhradně na dotykové ovládání prstem, bez využití stylusu, jak to bylo běžné u konkurence. Dnes se tento trend snaží kopírovat výrobci všech konkurenčních systémů. Překážkou rozšíření je (přinejmenším v České republice) vysoká cena přístroje (v prosinci 2009 se nedotovaný model iPhone 3GS s 16 GB pamětí dal u operátora Vodafone koupit za částku 16.777 Kč). Překážkou tvorby aplikací pro tento systém jsou omezení ze strany firmy Apple. Ta sice uvolnila *SDK (Software Development Kit)*, nicméně využívat jej lze jen na počítačích od Applu a jen po zaplacení ročního poplatku 99 nebo 300 amerických dolarů (soukromá / firemní vývojářská licence). Posledním problémem je šíření hotových aplikací – to je možná jen přes službu AppStore, jakýsi centrální bod pro distribuci, kde musí Apple veškeré vystavované aplikace nejdříve schválit. Ty poté mohou být nabízeny zdarma nebo za libovolnou cenu, ze které si Apple, jako odměnu za zprostředkování obchodu, 30 % strhává.

Dalším vyřazeným kandidátem je platforma *BlackBerry* severoamerické firmy *RIM (Research In Motion)*. I když podle grafu je celosvětově zastoupena poměrně hodně, v Evropě a především v České republice se prakticky nevyskytuje. Je to dáno jednak jejím zaměřením na korporátní klientelu a nutnou podporu ze strany operátora, který provozuje *BES (BlackBerry Enterprise Server)* nebo vlastnictvím odpovídající technické infrastruktury uvnitř firmy (což zahrnuje nákladné serverové operační systémy Microsoft Server, e-mailová řešení Microsoft Exchange nebo Lotus Domino a nákup licencí pro každý připojovaný BlackBerry přístroj). Tyto požadavky ostře kontrastují s trendy v opensource komunitě, ke které vyvíjená aplikace směřuje.

Podle grafu je nejrozšířenějším systémem Symbian. Ten byl v roce 1998 firmami Ericsson, Motorola, a Nokia představen jako následník systému *EPOC* z kapesních přístrojů *Psion*. Z počátku se jednalo o uzavřený systém, ale od roku 2008 jsou zdrojové kódy postupně uvolňovány pod

otevřenou licencí *EPL (Eclipse Public Licence)*. Z hlediska vývojáře je problematický především fakt, že systém existuje v mnoha edicích, které mezi sebou často nejsou zcela kompatibilní. Další překážkou je historický fakt, že Symbian byl původně vyvíjen pro telefony bez dotykového displeje, ten je podporován až v posledních dvou větvích *S60 R5 (Series 60, release 5)* a *UIQ (User Interface Quartz)*, kde druhá podle článku <http://www.esato.com/news/article.php?id=1811> v roce 2009 zbankrotovala. Podle největšího českého katalogu mobilních telefonů a smartphonů na serveru <http://www.mobilmania.cz> má dotykový displej pouze 19 přístrojů (9 s S60 R5 a 10 s UIQ) ze 108 dosud vyrobených (a evidovaných) modelů, tedy jen 18 %. Pokud bychom tuto skutečnost promítli do výše uvedeného grafu, připadlo by pro Symbianovské dotykové přístroje pouze necelých 9 % trhu. A kdybychom brali v potaz i vzájemnou nekompatibilitu uvedených dvou větví, byla by čísla ještě horší. Z těchto důvodů nepovažuji ani Symbian za vhodnou platformu pro vývoj požadované aplikace. Zbývá tedy operační systém Windows Mobile od Microsoftu.

## 2.4.3 Windows Mobile

Windows Mobile je uzavřený operační systém firmy Microsoft určený pro PDA a smartphony, který je založen na *Windows CE (Compact Edition, systém pro vestavěné systémy)*. Jeho vzhled je odvozený od klasických desktopových Windows, stejně tak nabízená funkcionality je v jistých mezích podobná – poskytuje určitou podmnožinu *Windows API*. První verze této řady operačních systémů, *Windows Mobile 2003* z ledna 2003 už podporovala telefonní funkce, dotykový displej i Bluetooth komunikaci. Další verze s označením 5, uvolněná v květnu 2005 přinesla centrální správu GPS přijímačů a podporu tzv. *landscape* režimu zobrazení („na šířku“). Poslední řada 6 (od února 2007) se zaměřuje především na smartphony a přidává funkcionality v této oblasti, což je pro nás vzhledem k požadavkům definovaným výše nezajímavé. V době psaní tohoto textu byla nejnovější dostupná verze 6.5.3 z února roku 2010. Od základů přepracovaná verze 7, označovaná jako *Windows Phone 7*, je plánována na druhou polovinu roku 2010.

Z hlediska programátora se jedná o celkem přívětivý systém a existuje několik jazyků, ve kterých lze pro Windows Mobile programovat. První možností je psát aplikace v C++ s využitím *Windows API*, na vyšší úrovni pak s využitím *Compact .NET Frameworku* nebo v Javě pro Java VM (*Virtual Machine*). Zmíněný *Compact .NET Framework* představuje podmnožinu plného, desktopového *.NET Frameworku* a tudíž se vývojář pracující s *.NET* technologií na desktopu nemusí učit kompletně novým postupům. Microsoft nabízí těmto programátorům kompletní SDK, které úzce spolupracuje s vývojovým prostředím Visual Studio a jazyky C# a Visual Basic. SDK nabízí i desktopový emulátor různých verzí Windows Mobile pro jednodušší testování aplikací. Mimo tyto kompilované jazyky existují i porty některých skriptovacích jazyků, například *PythonCE* pro Python.





Obrázek 6: referenční přístroj MIO P550 s Windows Mobile 5.  
převzato z <http://www.ia.ro/magazen/images/GPS%20PDA%20Mio%20P550.jpg>

Co se distribuce hotových aplikací týče, nabízí se zde také více možností. Aplikace ve formě instalačního .CAB balíčku může být pomocí synchronizační utility, přes Bluetooth, infraport nebo WiFi, z webu nebo přes paměťovou kartu nahrána do zařízení a odtud ručně nainstalována, dále může být šířena ve formě spustitelného instalátoru, který aplikaci z desktopu do zařízení pomocí synchronizační utility nahraje automaticky, nebo za třetí nově (od února 2009) i přes online službu *Windows Marketplace for Mobile* (obdoba Appstore od Apple).

Rozšíření a dostupnosti přístrojů s Windows Mobile jsem se věnoval na začátku kapitoly 2.4.2. Na základě uvedených skutečností tedy můžeme prohlásit, že tato platforma všem kladeným požadavkům vyhovuje.





Obrázek 7: externí GPS přijímač NAVILOCK BT-359 komunikující přes Bluetooth.

převzato z <http://p.gzhls.at/212834.jpg>

Jako referenční přístroj pro vývoj a testování bylo zvoleno PDA *MIO P550*, viz Obrázek 6. Disponuje procesorem s taktem 400 MHz, 64 MB RAM, dotykovým displejem s rozlišením 320 × 240 bodů (tzv. QVGA, čtvrtinové VGA), Bluetooth připojením i interním GPS přijímačem. Umožňuje tedy aplikaci testovat jak ve spolupráci s vestavěnou, tak s externí jednotkou – v našem případě *NAVILOCK BT-359* zobrazenou na Obrázek 7. Výkon tohoto přístroje je na dnešní dobu podprůměrný, například současné Windows Mobile 6.5 zařízení *HTC HD2* disponuje procesorem s frekvencí 1 GH a operační pamětí 512 MB. Tento fakt sice klade větší nároky na optimalizace programu pro výkon, na druhé straně nám dává jistou záruku toho, že software fungující na referenčním kusu nebude mít výkonnostní problémy na jiných (novějších) strojích.

## 3 Existující řešení

Idea využití mobilní výpočetní techniky coby digitálního geodetického zápisníku není sama o sobě nikterak novátorská. Pro zvolenou platformu Windows Mobile je dostupných několik komerčních programů sloužících uvedenému účelu. Patrně nejvěhlasnějším profesionálním produktem je *ArcPad* od dříve vzpomínané společnosti ESRI. Jeho vlastnostmi, představením kladů a záporů se zabývá tato kapitola. Poté diskutuje opodstatnění vývoje vlastního softwarového produktu a ukazuje výhody, které nový produkt uživatelům přinese.

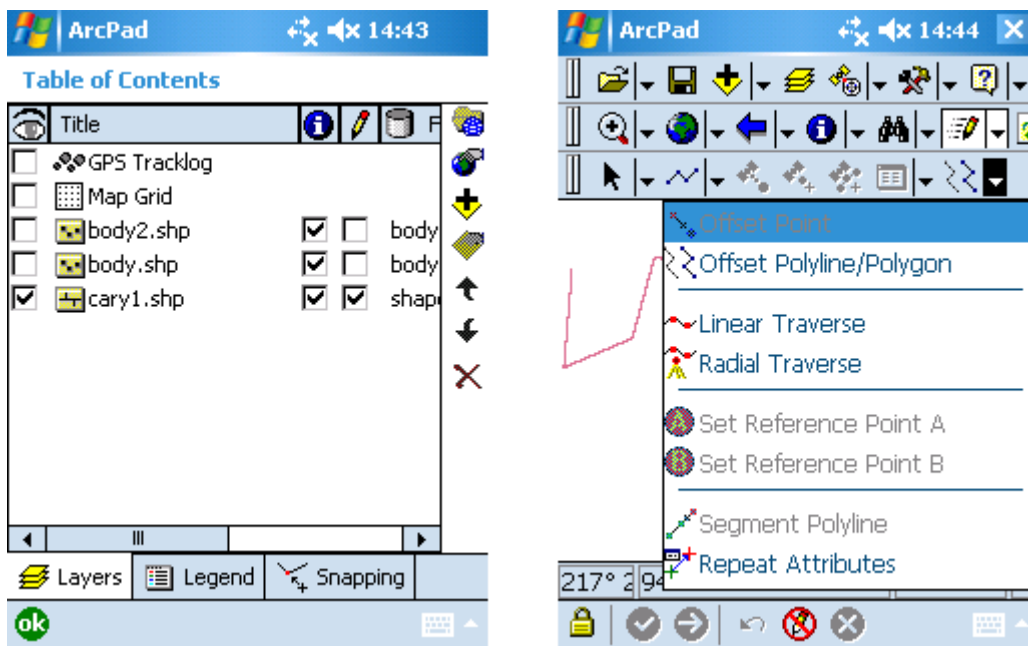
### 3.1 ArcPad

Jedná se o softwarový balík pro přístroje s Windows Mobile 5 a novějšími. Nároky na hardware výrobce neuvádí, nicméně na referenčním přístroji program fungoval uspokojivě. Aplikace je na datové úrovni úzce provázána s desktopovým systémem ArcGIS, což je vzhledem ke stejnému výrobcí pochopitelné. Problém to představuje pro uživatele, kteří by chtěli ArcPad plně využívat v kombinaci s jiným programem. Na úrovni vektorových vrstev by komplikace nastat neměly, protože výchozím formátem je standardizovaný shapefile. Problém ale nastává u vrstev rastrových, kde je nasazen vlastní, uzavřený formát `.apg` či `.aph` pro fotomapy. Přímé načtení standardního rastrového obrazu ArcPad vůbec nepodporuje, což vnímám jako zásadní nedostatek.

Kromě základních funkcí digitálního zápisníku poskytuje možnost připojení ke geografickým databázím přes internet, spolupráci nejen s GPS přijímačem, ale i dalšími periferiemi jako jsou dálkoměry nebo digitální fotoaparáty. Rozšířená je i paleta editačních funkcí, kde je uživateli umožněno vkládat předdefinované tvary jako mnohoúhelníky, kružnice, elipsy, objekty kreslené „od ruky“ nebo dokonce text. Protože ale shapefile ukládání těchto tvarů nepodporuje, jsou převedeny na množství linií, což velmi komplikuje jejich pozdější editaci a zvětšuje objem datových souborů. Ke standardním manipulačním funkcím přidává interaktivní editaci objektů. To znamená, že jednotlivé vrcholy mohou být přesunovány přímo pomocí dotykového pera přes displej. Vzhledem k nízkému rozlišení obrazovky může tento způsob v závislosti na použitém měřítku zobrazení vnést do dat neakceptovatelně velkou chybu. Dalšími možnostmi je vytváření bufferů (ekvidistant) zvolených prvků, posun celých objektů nebo jejich rotace. Nechybí ani některé analytické funkce jako zjišťování délek liniových segmentů, výpočet vzdáleností objektů nebo obsahu polygonů. Stále ale musíme mít na paměti, že se jedná o program pro kapesní počítač s malým displejem, který bude používán ve venkovních podmínkách, takže zůstává otázkou, nakolik uživatel tyto funkce, typické spíše pro desktop, ocení a zda je vůbec dokáže využít.

Důsledkem takto široké nabídky funkcí je také poněkud přebujelé uživatelské rozhraní, kde na QVGA displeji  $240 \times 320$  bodů zůstává pro mapový výřez pouze 56 % plochy, viz Obrázek 8. Nejen

že plocha pro vlastní práci je neúměrně zmenšena, ale v rozsáhlém menu se i špatně orientuje. Horní nástrojový panel má tři řady s 21 ikonami, které v některých případech aktivují vysouvací menu s až 14 položkami. Nad panelem je systémová nástrojová lišta, která zabírá 7 % plochy, avšak žádné ovládací prvky související s mapováním neobsahuje a je tedy pro uživatele bezcenná. Další lišta s kontextovými ikonami a se stavovým řádkem je u spodní hrany obrazovky.



Obrázek 8: snímek obrazovky přehledu vrstev a hlavní obrazovky s aktivovaným podmenu programu ArcPad 7.

Uživateli se nabízí i široká množina parametrů, které si může přizpůsobit. Nejvíce viditelné je to u konfigurace vykreslování jednotlivých vrstev – lze nastavit symboliku použitou pro body, tloušťky, barvy a styly u linií a u polygonů barvy a styly jak obrysu, tak i výplně. Nad vykreslené vrstvy je možné přidat směrovou růžici nebo grafické znázornění měřítka. K dispozici jsou také dva vnitřní skriptovací jazyky založené na VBScriptu a JScriptu. Slouží pro tvorbu maker a automatizaci často prováděných sekvencí kroků.

Pro menší firmy a jednotlivce může být problém i cena jednotlivých licencí. Na domovském webu společnosti se v květnu 2010 prodávala jedna instalace za 700 amerických dolarů, české zastoupení ceny nezveřejňuje. Při přímém přepočtu s dolarovým kurzem 20,50 Kč to dělá 14.350 Kč. Skutečná koncová cena bude ale vyšší minimálně o clo, DPH a marži prodejce – při pohledu na rozdíly cen jiných softwarových produktů (například sady Microsoft Office nebo Adobe Creative Suite) může být až dvojnásobná oproti pouhému přepočtu měny.

## 3.2 Opodstatněnost další aplikace

V kontextu těchto skutečností se může zdát, že konkurovat takovému komplexnímu produktu je nemožné. Z hlediska množství nabízených funkcí tomu tak pravděpodobně bude. Je potřeba si však uvědomit, že na tvorbě představeného produktu pracuje celý vývojový tým mnoho let, což znamená, že do vývoje bylo investováno mnohem více prostředků a času, než je v jedné diplomové práci během omezené doby možné. Navíc lze předpokládat, že ne všichni uživatelé takto bohatou nabídku funkcí potřebují nebo vůbec chtějí. Čím více možností program nabízí, tím více systémových prostředků spotřebovává a o to je náročnější na údržbu jak z pohledu tvůrce, tak uživatele. Každý řádek kódu, který je naprogramován musí být následně testován, debugován, dokumentován a podporován. Proto se ukazuje jako nejlepší nepotřebné řádky vůbec nepsat. V duchu této myšlenky se nese obsah knihy [8] *Getting Real* popisující agilní vývojářské praktiky a postupy uvnitř internetové společnosti [37signals.com](http://37signals.com). Autoři tvrdí, že u softwarových projektů je důležité izolovat minimální nezbytné množství funkcionality a tu naprogramovat. Zkracuje se tím doba vývoje a zjednodušuje udržitelnost kódu. Pokud budou uživatelé další funkce opravdu potřebovat, mohou je tvůrci doprogramovat později. Důležité je mít použitelný produkt k dispozici rychle. Kniha sice popisuje vývoj webových aplikací, ale klíčové myšlenky lze uplatnit i v jiných oblastech softwarového inženýrství. Závěr je takový, že množství nabízených funkcí není jediným kritériem úspěšnosti softwarového produktu.

Na jakou skupinu uživatelů je tedy nový produkt zaměřen a co jim přinese? Cílovou skupinou jsou jednotlivci a malé firmy, pro které zpracovávání geodat nemusí být hlavní pracovní činností. Těm nemusí vadit menší množství funkcí programu, které vyváží cena – produkt totiž bude šířen volně, včetně zdrojových kódů. Do budoucího vývoje se tak mohou zapojit i další zájemci. Na publikaci opensource GIS projektů se specializují například webové katalogy <http://opensourcegis.org> nebo <http://freegis.org>. Jelikož se jedná o mezinárodní projekty, bude uživatelské rozhraní komunikovat v angličtině. Nemyslím, že pro potenciální české uživatele to bude představovat problém, protože textu nebude mnoho a bude se jednat o popisky prvků uživatelského rozhraní nebo krátká technická sdělení a chybová hlášení.

Cena však nebude jedinou výhodou – oproti ArcPadu bude program disponovat odlehčeným a mnohem jednodušším uživatelským rozhraním, což pro uživatele bude znamenat více místa na displeji a tím pádem i více komfortu při práci.

Celkové množství funkcí sice bude nižší, ale minimálně jedna zcela nová a užitečná vlastnost přibude – přímé načítání rastrových obrazových souborů pro použití v roli podkladové fotomapy, které v ArcPadu z neznámých příčin chybí.

Výše prezentované skutečnosti jsem vyhodnotil tak, že vývoj aplikace má smysl. Jejím návrhem, realizací a praktickým vyzkoušením se budou zabývat další kapitoly.

## 4 Návrh aplikace

Čtvrtá kapitola se věnuje celkovému návrhu programu. Nejdříve sumarizuje požadavky vzešlé z předchozích dvou kapitol a dále se zabývá volbou vývojových prostředků. Vybrán byl jazyk C# ve spolupráci s Compact .NET Frameworkem, jako integrované vývojové prostředí Microsoft Visual Studio 2008, které mimo jiné poskytuje emulátor mobilního zařízení. Protože aplikace nebude uzavřená, ale bude komunikovat s jinými programy a systémy, je další podkapitola věnována výběru souborových formátů k ukládání a načítání dat. Pro vektorová data to je standardizovaný shapefile, pro fotomapy několik typů běžných rastrových obrazových formátů. Nakonec je popsána struktura a základní funkce uživatelského rozhraní – rozložení ovládacích prvků jednotlivých obrazovek včetně jejich návazností.

### 4.1 Upřesnění požadavků

Na základě poznatků z předchozích kapitol můžeme nyní definovat konkrétní požadavky na výslednou aplikaci. V bodech to jsou:

- Program musí být spustitelný na PDA nebo smartphone s operačním systémem Windows Mobile 5 nebo novějším. Tato verze byla vydána v roce 2005, nejedná se tedy o žádnou novinku, na druhé straně je díky tomu systém dostatečně rozšířený a novější WM řady 6 jsou zpětně kompatibilní.
- Musí reagovat na ovládání pomocí dotykového displeje. Intuitivní práce bude podpořena přehledným a současně co možná nejvíce minimalistickým uživatelským rozhraním, aby zbylo více zobrazovací plochy pro vlastní práci.
- Informace o geografické poloze bude přednostně získávat z GPS přijímače prostřednictvím standardizovaného protokolu NMEA. Nutná je možnost komunikace jak s interní, tak s externí jednotkou připojenou přes Bluetooth. Pro případ potřeby ruční korekce dat nebo jako záloha v případě nedostupnosti GPS signálu však bude možné vkládat údaje o poloze vhodným způsobem i ručně.
- Bude podporovat práci s vrstvami a to jak vektorovými, tak rastrovými ve formě grafického podkladu – fotomapy. Vrstev může být obecně libovolný počet a může jich být zobrazeno více současně. Pro editaci bude aktivní vždy právě jedna vrstva.
- Ve vektorových vrstvách bude umožněna práce s jednotlivými body, úsečkami a polygony nebo jejich částmi. U bodů bude možné aktualizovat jejich souřadnice, linie a polygony půjde zpřesňovat i zjednodušovat, tedy přidávat i odebírat vrcholy. Pro

veškeré typy objektů bude možné prohlížet a editovat ne-geografické atributy s nimi spojené. Bude možné celé objekty mazat.

- Po provedení editačních operací bude uživateli umožněno vracet se k předchozím stavům objektů – funkce „krok zpět“.
- Bude možné vytvářet nové objekty všech tří podporovaných prostorových typů.
- Protože se jedná primárně o aplikaci pro sběr dat v terénu, nejsou požadovány analytické a vizualizační funkce, další zpracování tohoto druhu se předpokládá na desktopu.
- Naopak velký důraz bude kladen na možnosti práce s mapovým výřezem. Tím se rozumí posouvání okna nad mapou a různé varianty oddalování a přibližování pohledu (skokové, na vybranou oblast, na vybraný objekt, na aktivní vrstvu, na vše). Bude možné vracet se k předchozím pohledům a definovat vlastní pojmenované pohledové záložky.
- Pro následné zpracování dat na desktopu bude program umožňovat ukládání a přenos získaných dat ve standardizovaných formátech.
- Uživatel bude moci zvolit, s jakým souřadným systémem hodlá pracovat. Jako dostatečné minimum bude sloužit světový standard *WGS-84 (World Geodetic System)* ustanovený roky 1984 v USA), který je výchozím systémem pro GPS, a *S-JTSK (Souřadnicový systém Jednotné trigonometrické sítě katastrální)*, který je rozšířený v České republice.

## 4.2 Vývojové prostředky

Jako programovací jazyk byl vybrán C#. Jde o vysokoúrovňový objektově orientovaný jazyk vyvinutý rovněž firmou Microsoft. První verze byla vydána v roce 2002, aktuální (třetí) v 2007 a v současnosti je vyvíjena verze čtvrtá. C# byl schválen jako standard organizacemi ECMA i ISO. Jazyk má několik charakteristických vlastností: syntaxe vychází z jazyka C, neexistuje zde vícenásobná dědičnost, každá třída tedy může být potomkem maximálně jedné jiné, na druhou stranu může implementovat více rozhraní. Neexistují žádné globální proměnné ani funkce či procedury – všechny atributy a metody musí být deklarovány uvnitř tříd. Nepracuje se s implicitními typovými konverzemi (například `integer` na `boolean`). Rozlišují se velká a malá písmena, jde tedy o *case sensitive* jazyk.



Obrázek 9: snímek okna emulátoru Windows Mobile 5.

S tímto jazykem se úzce pojí *.NET platforma* („dot NET“) taktéž od Microsoftu, jejíž základní součástí je *.NET Framework* – podpůrné knihovny a běhové prostředí. Nejčastěji používaným jazykem pro *.NET* platformu je právě C#, ale není to podmínkou – podporována je spousta dalších, například funkcionální *F#*, pascalovské *Delphi*, skriptovací *IronPython* a další. Variabilita a přenositelnost je umožněna díky mezivrstvě *CIL (Common Intermediate Language)*, jakési obdoby *Java Bytecode*, do které jsou veškeré zdrojové kódy překládány. Konečný převod do strojového kódu a vlastní běh aplikace na MS Windows má na starosti běhové prostředí *CLR (Common Language Runtime)*, rovněž součást *.NET* frameworku. Existuje i *CIL* implementace pro jiné operační systémy než MS Windows – například open source *Mono* pro Unixové systémy vyvíjené firmou Novell.

Pro nás zajímavý je odlehčený *Compact .NETFramework* (dále jen CNF) určený pro vývoj mobilních aplikací na PDA a smartphonech (zajímavostí je, že *.NET CF* podporuje i herní konzole *Xbox 360*). Obsahuje jak podmnožinu funkcí z plného frameworku, tak některá rozšíření využitelná právě na mobilních zařízeních (specifické prvky uživatelského rozhraní a dotykového ovládání). Aktuální verze 3.5 pochází z ledna 2008 a je kompatibilní s Windows Mobile 2003 SE a novějšími. Z omezení oproti desktopové verzi vyvíjenou aplikaci ovlivní užší nabídka vestavěných tříd s omezeným počtem dostupných metod a vlastností. Redukována je i sada poskytovaných vizuálních komponent pro tvorbu uživatelského rozhraní, na druhou stranu přibývají některé specifické události.

Nejvíce omezena je paleta tříd a metod pro práci s grafikou. Konkrétní problémy a jejich řešení budou probrány v kapitole věnující se implementaci.

Vývoj aplikací pro .NET CF v jazyce C# je možný v integrovaném vývojovém prostředí (*IDE – integrated development environment*) Microsoft Visual Studio 2008. Toto propracované prostředí s historií od roku 1997 nabízí několik nástrojů: základním je editor kódu s barevným zvýrazňováním syntaxe a návrhy na dokončení rozepsaných identifikátorů a konstrukcí jazyka – tzv. *intellisense*. Pro ladění je zde debugger umožňující nastavování *breakpointů* ve zdrojovém kódu, krokování nebo sledování obsahu vybraných proměnných. Dále to je sada několika vizuálních návrhářů umožňujících tvorbu uživatelského rozhraní z předpřipravených komponent, design tříd v souladu s UML nebo návrh databázového schématu. Podpora refaktorisace kódu spočívá ve funkcích globálního přejmenovávání identifikátorů, změně pořadí parametrů metod nebo třeba extrakci metody. Tyto hlavní součásti jsou doplněny dalšími nástroji, jako je podpora týmové spolupráce, databázová konzola a mnohé další.

S Visual Studií úzce spolupracuje i SDK pro Windows Mobile, který obsahuje emulátory různých verzí Windows Mobile spolu s různými rozlišeními displejů virtuálních zařízení. Tato vlastnost je velmi praktická, neboť dnes jsou na trhu zařízení s rozlišením od  $240 \times 320$  po  $480 \times 800$  pixelů, včetně různých poměrů stran obrazovky s orientací jak na výšku, tak na šířku. Dále je možné emulátor propojit se síťovou kartou nebo sériovým portem hostitelského počítače, což umožňuje pracovat s externím GPS modulem připojeným přes Bluetooth. Vzhled emulátoru operačního systému Windows Mobile 5 s displejem  $240 \times 320$  pixelů ukazuje Obrázek 9.

## 4.3 Souborové formáty

### 4.3.1 Rastry

Aby bylo možná rastrový obrazový soubor (například letecký snímek) v GIS aplikacích smysluplně používat, musí být takzvaně *georeferencován*. To znamená, že musí být pomocí metadat jednoznačně určeno, jak zobrazená oblast zapadá do použitého souřadného systému. Typicky je tato informace reprezentována šesti desetinnými čísly: dvě souřadnice levého horního pixelu, šířka a výška jednotlivých pixelů v souřadných jednotkách a případně pootočení rastru podle souřadných os. Při georeferencování bývá obraz často podle potřeby přetočen rovnou, takže parametry otočení v metadatech bývají nulové, což jednoznačně usnadňuje další práci a šetří systémové prostředky, protože se jedná o operaci výpočetně poměrně náročnou. Vzhledem k požadavkům procesu georeferencování z hlediska obsluhy (potřeba velkého monitoru a přesného vstupního zařízení) budeme dále předpokládat, že aplikace bude operovat s již zpracovanými obrazy. Vlastní postup georeferencování se tak pro nás stává nezajímavým.



Z technického hlediska mohou být metadata přímo součástí obrazového souboru (typicky v hlavičce), nebo mohou být uložena zvlášť. První přístup se uplatňuje u *GeoTIFF* formátu pro obrazy *TIFF (Tagged Image File Format)*, druhý je univerzální pro prakticky libovolný obrazový formát, který je doprovázen malým textovým souborem, takzvaným *world file*, typicky s příponou *.twf* (například obraz *snimek.jpg* a k němu metadata *snimek.twf*).

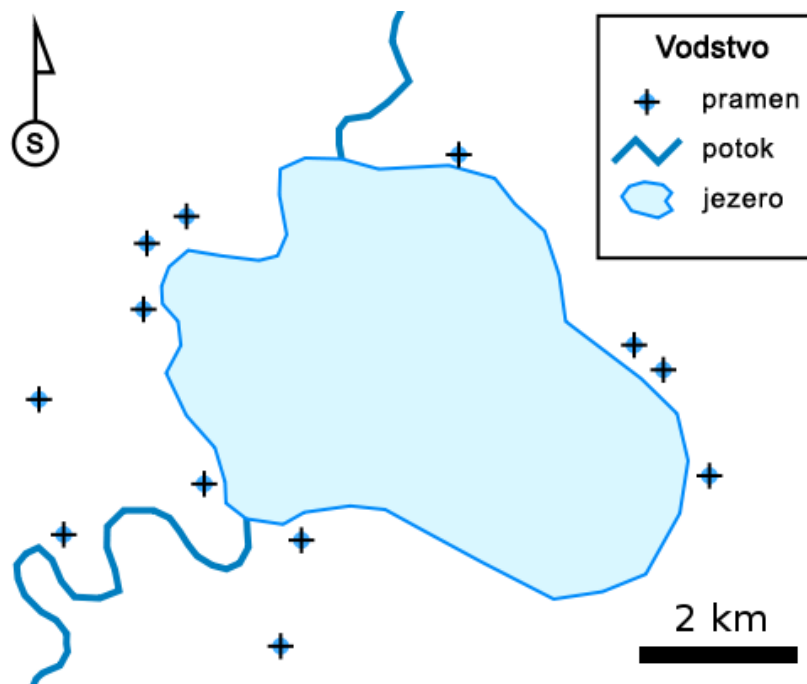
CNF bohužel nepodporuje práci s TIFF grafikou, tudíž bude program pracovat s georeferenčními daty uloženými v samostatném souboru. Uživatelé s mapami ve formátu GeoTIFF mohou v desktopové aplikaci provést export metadat a TIFF konvertovat do některého z podporovaných grafických formátů – JPG, GIF nebo PNG, který je hojně využíván například v GRASSu. Užitečná hodnota programu se pro ně tedy nesnižuje.

### 4.3.2 Vektory

Rozšířeným a otevřeným vektorovým formátem, který podporuje i GRASS, je takzvaný *shapefile*. Specifikaci formátu počátkem devadesátých let 20. století vyvinula a dále udržuje již dříve zmíněná společnost ESRI. Ačkoliv GIS nástroje této firmy jsou komerční, specifikace formátu je všem dostupná zdarma v dokumentu [9]. Důvodem uvolnění specifikace je pravděpodobně snaha zachovat datovou kompatibilitu mezi vlastními a konkurenčními programy.

Jedná se tedy o vektorový formát, který umožňuje ukládání prostorových dat pomocí tří základních entit: bodů, linií a polygonů. Od každého z těchto objektů existují další dva odvozené podtypy pro trojrozměrná data. V souladu s myšlenou z úvodní kapitoly, že práce s trojrozměrnými daty není v GIS praxi zcela typická, budete vyvíjený program v počáteční verzi podporovat pouze základní entity. Podpora pro manipulaci s rozšířenými typy může být doprogramována později.

Nutno podotknout, že liniemi jsou zde myšleny úsečky, nikoliv křivky, se kterými *shapefile* pracovat neumí. Podle specifikace může jedna vrstva obsahovat pouze jeden typ entit (ve specifikaci se o entitách hovoří jako o *features*), není tedy možné v jednom souboru ukládat současně například body a linie. Na první pohled je to omezující podmínka, která má za následek nárůst počtu souborů, protože pro uložení logicky souvisejících entit je potřeba více oddělených vrstev. Například pro uložení mapové kompozice při zkoumání vodstva v určité oblasti (viz Obrázek 10) musíme použít tři vrstvy místo jedné – zvlášť pro vodní zdroje (body), toky (linie) a plochy (polygony). Na druhé straně tato vlastnost usnadňuje práci programátorovi – protože může očekávat, že v souboru jsou pouze entity homogenního typu, nemusí pro každou z nich volit jiný způsob práce, může s nimi manipulovat hromadně (a snad i rychleji) při transformacích, načítání a ukládání do paměti. Z pohledu uživatele aplikace je manipulace s tímto souborovým typem naopak o něco pomalejší – musí častěji přepínat, která vrstva je aktivní pro editaci. Pokud by chtěl pracovat se smíšeným obsahem, musel by využít datový formát a software, který taková data podporuje, soubory do něj importovat a sloučit. Vyvíjená aplikace se bude držet oficiální specifikace a tento přístup podporovat nebude.



Obrázek 10: mapová kompozice vodstva sestávající ze tří vrstev – body, linie a polygony.

Další vlastností shapefile je, že ukládá pouze informace o geometrii, nikoliv o topologii. V terminologii z úvodní kapitoly se jedná o špagetový model.

Ačkoliv to z názvu přímo nevyplývá, je u tohoto datového formátu ve skutečnosti jedna vrstva fyzicky uložena ve více diskových souborech. Nezbytně nutné jsou tři, ale může jich být podle potřeby i více.

Soubor s koncovkou `.shp` (*shape format*), obsahuje vlastní geometrii. Struktura souboru je tvořena hlavičkou pevné délky 100 bajtů a následnými záznamy proměnlivé velikosti pro jednotlivé entity. Každá taková entita má svou 8 bajtovou hlavičku (obsahující pořadové číslo a délku záznamu) a dále vlastní data. Ta začínají 4 bajtovým typem záznamu a pokračují daty specifickými pro jednotlivé typy entit. Schematicky strukturu ukazuje Obrázek 11, přesné významy jednotlivých bajtů jsou pak popsány v dokumentaci [9]. Údaje o poloze bodů se ukládají jako dvě desetinná čísla `double` vyjadřující x a y souřadnice. Interpretace těchto čísel závisí na použitém souřadném systému. Nejčastěji uložené hodnoty vyjadřují zeměpisnou šířku a délku s touto sémantikou: x odpovídá délce, kde kladné hodnoty značí východní délku a y šířce, s kladnými hodnotami pro severní šířku.

Hlavička souboru	
Hlavička entity	Data entity
Hlavička entity	Data entity
...	
...	
Hlavička entity	Data entity

Obrázek 11: základní struktura .shp souboru s geometrií.

Koncovka `.shx` (*shape index format*) je vyhrazena pro indexový soubor jednotlivých entit obsažených v geometrickém souboru a využívá se pro rychlý náhodný přístup k entitám. Obsah je tvořen stejnou, 100 bajtovou hlavičkou jako u `.shp`, která je následována fixními, 8 bajtovými záznamy. Ty se skládají z offsetu položky od začátku souboru a délky této položky. Právě díky položkám pevné délky je navigace v tomto souboru (na rozdíl od `.shp`) snadná i rychlá.

Poslední, dle specifikace nezbytný soubor má koncovku `.dbf` (*atribute format*). Obsahuje tabulku atributů vztahujících se k entitám z geometrického souboru. Data jsou uložena ve standardizovaném formátu *dBase III*, jehož specifikace je dostupná na webu [10]. Tento formát má některá omezení, se kterými musíme dopředu počítat: umožňuje uchovávat maximálně 255 atributů (sloupců) s názvy délky nejvýše 10 znaků. Možné datové typy jsou celé číslo, desetinné číslo s pevným, předem daným počtem desetinných míst, řetězec o maximální délce 254 znaků a datum od 1. 1. 1900 dále.

Specifikace dále říká, že názvy souborů by měly být tvořeny podle klasického schématu 8.3 – název až 8 znaků, přípona 3 znaky. Vzájemná sounáležitost souborů se určuje na základě shodné části před tečkou, takže například základní sada souborů pro vrstvu čerpacích stanic by mohla být tvořena soubory `benzinky.shp` (jednotlivé body = čerpací stanice), `benzinky.shx` (index) a `benzinky.dbf` (atributy, například značka stanice, otevírací doba nebo poštovní adresa). Počet záznamů ve všech souvisejících souborech se musí shodovat a jednotlivé záznamy si, co se týče pořadí, musí odpovídat.

### 4.3.3 Projektové a konfigurační soubory

Kromě vektorových a rastrových vrstev bude program muset umět pracovat i s dalšími daty vztahujícími se k mapovým projektům. Pro jednotlivé instance bude nutné zaznamenat použitý souřadný systém, naposledy nastavené měřítko přiblížení a zobrazovaný výřezem mapy, odkazy na soubory s daty vrstev, údaje o jejich viditelnosti a výběru pro editaci, georeferenční data případného grafického podkladu a nastavení pohledových záložek.

Vedle těchto projektových informací bude existovat ještě jeden konfigurační soubor k samotnému programu. V něm budou uchovávány parametry GPS přijímače, především číslo sériového portu a přenosová rychlost komunikace plus další údaje nezbytné pro fungování programu a uživatelské přizpůsobení grafického rozhraní aplikace.

Ani zde nemá smysl vymýšlet vlastní způsob uložení, k použití se nabízí všeobecně známý značkovací formát *XML (Extensible Markup Language)*. Jistý problém tohoto datového formátu je nevýhodný poměr délky vlastních užitečných dat k objemu značkovací rezie okolo. V případě naší aplikace ale nebude množství uchovávaných informací nijak velké, předpokládají se řádově stovky bajtů, takže tuto vlastnost můžeme ignorovat. Bohatě ji převáží snadná práce s daty, neboť podpora pro načítání, zpracování i ukládání XML souborů je jazyce C# přímo zabudována.

## 4.4 Uživatelské rozhraní

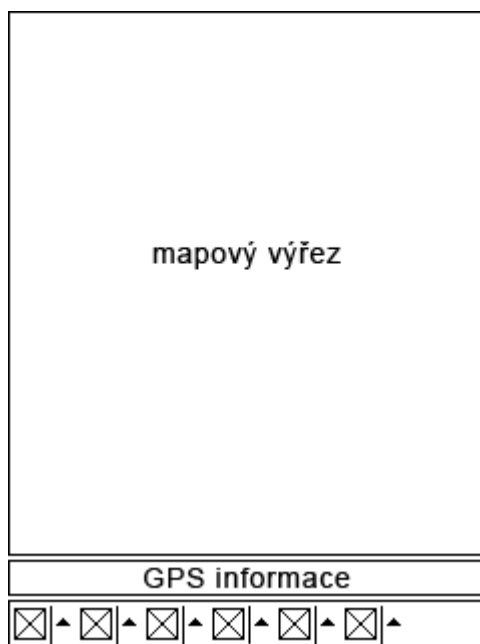
Jak bylo řečeno ve třetí kapitole, účinné uživatelské rozhraní bude jednou z předností vyvíjeného programu. Základními principy bude co největší užitná plocha, co nejmenší možná hloubka menu a v rámci možností i co nejmenší počet položek v nich. Redukce počtu záznamů lze dosáhnou sdružování více funkcí pod jedno tlačítko v závislosti na aktuálním stavu aplikace nebo probíhající operaci. Na druhé straně tento přístup může být pro uživatele matoucí, takže je potřeba slučování provádět opatrně.

Grafická podoba rozhraní bude sestávat ze standardních komponent, jako jsou klasická tlačítka, textová menu, vstupní pole, zaškrtačací pole nebo nástrojové lišty, které jsou poskytovány přímo CNF a lze s nimi v prostředí Visual Studia pracovat pomocí myši. Jelikož plocha displeje je malá, bude většina oken řešena jako celoobrazovkové, pouze potvrzovací dialogy nebo dotazy na vstup jednotlivých hodnot budou ve formě menších plovoucích dialogových oken.

Ve Windows Mobile aplikacích je u všech oken standardně vyhrazen horní řádek coby jakási obdoba taskbaru z desktopových Windows. Obsahuje křížek pro minimalizaci aplikace a ikonu pro přístup do systémového Start menu. Chování ikony křížku je ve srovnání s desktopovým systémem z počátku matoucí – ve výchozím nastavení totiž nezpůsobí uzavření aplikace, ale pouze její minimalizaci. Při dalším spuštění sice aplikace startuje rychleji, ale zase stále zabírá místo v operační paměti. Po delší práci se zařízením volné paměti ubývá a chod celého systému se zpomaluje. Protože některé aplikace možnost opravdového ukončení vůbec nenabízejí, musí být uzavřeny přes správce procesů nebo restartem systému. Protože ale tato lišta nenesie kromě tohoto tlačítka žádné ovládací prvky využitelné v aplikaci, bude z důvodů šetření místa v hlavním okně programu deaktivována. V dalších pomocných oknech pak bude obsahovat ikonu pro jejich skutečné zavření.

Řádek u spodní hrany displeje je vyhrazen buď pro textové roletové menu (dále jen textové menu), nebo nástrojový panel s grafickými ikonami. Protože ikony zabírají méně místa než odpovídající slovní reprezentace, bude v hlavním okně použita grafická varianta a v ostatních textová

bez vnořených položek. Textové menu bude typicky obsahovat pouze dvě položky „OK“ a „Storno“, čímž se na formuláři uvolní místo, které by tato standardní dialogová tlačítka zabírala. V obou případech je nutné počítat se systémovou ikonou pro aktivaci softwarové klávesnice, kterou Windows na řádek automaticky doplňuje. Z toho důvodu se na dolní panel základní šířky 240 pixelů vejdou přibližně tři až čtyři textové položky nebo šest ikon velikosti  $16 \times 16$  pixelů (včetně symbolu šipky indikujícího přítomnost rozbalovacího podmenu). Každá ikona bude představovat určitou skupinu souvisejících akcí, kde nejčastěji používaná funkce ze skupiny bude aktivována kliknutím na ni. Další operace ze skupiny budou přístupné z vnořeného textového menu, které se zobrazí právě po kliknutí na symbol šipky. Rozdělení funkcí do skupin a především volba „nejčastěji používané funkce“ je jedním ze zásadních bodů návrhu rozhraní. Zde bude důležitá role praktického ověření aplikace v terénu, na základě které může dojít ke korekci rozložení.



Obrázek 12: drátěný model rozložení prvků hlavního okna.

V hlavním okně se nad spodní lištou bude nacházet textový stavový řádek, kde bude zobrazována informace o poloze, případně další údaje získané z GPS přijímače. Zbylá část okna nad těmito ovládacími prvky bude celá věnována mapovému výřezu, kde se zobrazuje obsah mapových vrstev a probíhá manipulace s entitami. Barevnost zobrazení entit v odlišných vrstvách bude možné v programu odlišit, takže na první pohled bude například patrné, zda linie patří do vrstvy silnic nebo potoků. Díky popsané redukci rozhraní se mapová část okna oproti uspořádání z ArcPadu zvětší na 85 % plochy displeje. To oproti původním 56 % představuje podstatný a nezanedbatelný nárůst. Drátěný model rozložení prvků ukazuje Obrázek 12.

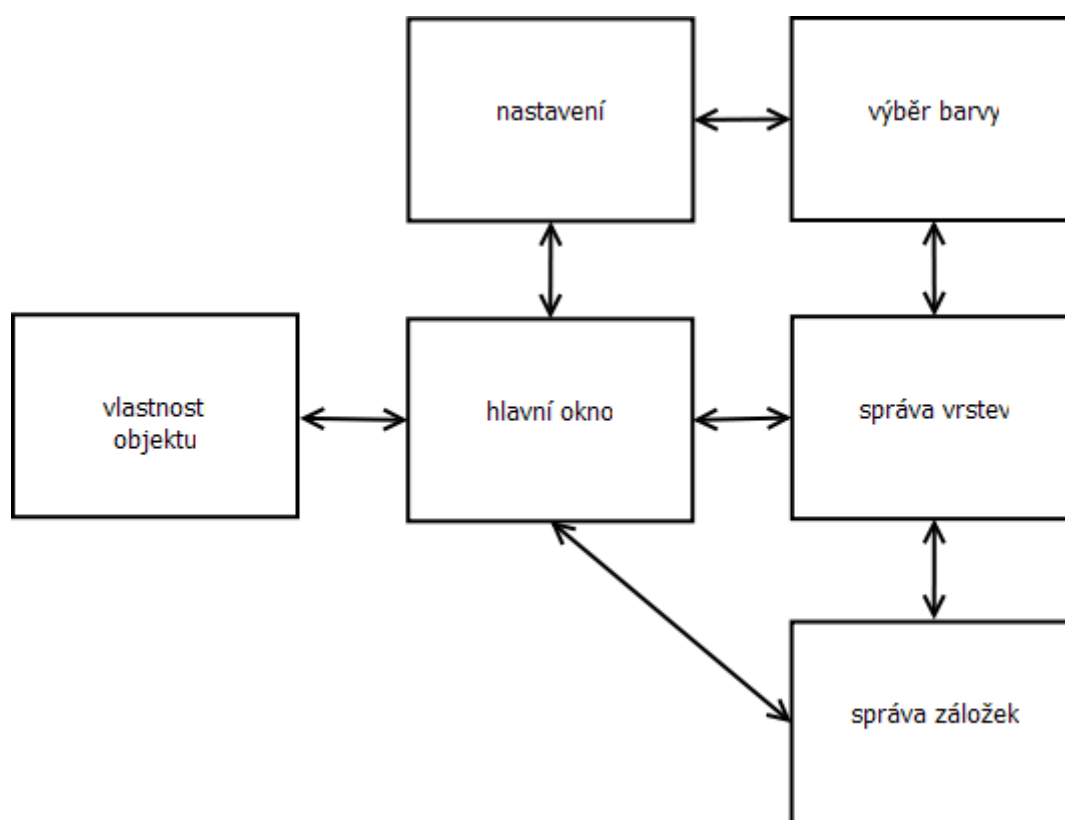
Hierarchická struktura menu bude následující (funkce na první úrovni bude aktivována přes ikonu, ostatní přes vysouvací textové menu):

- Uložení projektu
  - Vytvoření nového projektu
  - Načtení projektu z disku
  - Uložení projektu pod jiným jménem
- Správa vrstev
  - Nastavení programu
- Přiblížení
  - Oddálení
  - Přiblížení na veškerý obsah
  - Přiblížení na obsah aktivní vrstvy
  - Přiblížení na vybraný objekt
  - Předchozí pohled
  - Vytvoření záložky
  - Seznam záložek
    - Jméno záložky 1
    - Jméno záložky 2
    - ...
- Posun výřezu
  - Výběr vrcholu
  - Výběr liniového segmentu
  - Vlastnosti vybraného objektu
- Přidání vrcholu
  - Přesun vrcholu
  - Smazání vrcholu
  - Smazání objektu
  - Krok zpět
  - Dokončení nového objektu
- Centrování pohledu na aktuální GPS pozici
  - Ukázat podrobné informace o pozici
  - Vypnout / zapnout GPS přijímač

Kontextově se bude nejvíce měnit funkce ikony „Přidání vrcholu“. V režimu editace to bude pro liniovou vrstvu buď vložení vrcholu do segmentu (je-li nějaký vybrán) nebo prodloužení linie – je-li vybrán koncový bod, u polygonů půjde o vkládání vrcholu do vybraného segmentu. V režimu tvorby nového objektu způsobí akce okamžité vložení nového objektu, u linií a polygonů se při každém stisku přidá další vrchol do dočasného objektu, který se uzavře a uloží po aktivaci

„Dokončení nového objektu“. Další ikonou s dvěma režimy práce je přiblížení – při pouhém ťuknutí na obrazovku dojde ke skokovému přiblížení, ale při dotyku, následném tažení a oddálení pera dojde k přiblížení na vybranou oblast – obdélník mezi bodem dotyku a oddálení.

Kromě hlavního okna bude program obsahovat několik dalších obrazovek: okno vlastností vybraného objektu, konfigurační formulář aplikace, editace záložky, výběr barvy a správu vrstev. Pro okno výběru barvy bohužel není možné použít standardní systémový dialog známý z desktopových Windows – v CNF žádný takový k dispozici není. Je proto nezbytné naprogramovat vlastní. Bude obsahovat tři posuvníky pro úpravu jednotlivých barevných složek (červená, zelená a modrá), náhled namíchaného odstínu a paletu předpřipravených barev. Nejsložitější z těchto formulářů bude sloužit pro správu vrstev a dalších údajů souvisejících s otevřeným projektem. Protože zobrazovaných údajů už bude poměrně hodně, bude okno dále děleno pomocí záložek na několik sekcí: vektorové vrstvy, rastrový podklad, volba souřadného systému a správa pohledových záložek. V posledně uvedené podskupině bude uveden seznam již definovaných pohledů, u nichž bude možné měnit pojmenování a pořadí v seznamu nebo je úplně odstranit. Pro změnu jména bude sloužit samostatný formulář, jenž se využije i při definici nového pohledu – přes funkci dostupnou z vnořeného menu na hlavní obrazovce. Zbylé dva formuláře pro nastavení parametrů a editaci vlastností objektu žádné zajímavosti neskrývají. Obrázek 13 ukazuje, jak na sebe budou jednotlivá okna navazovat.



Obrázek 13: návaznost obrazovek aplikace.

## 5 Implementace

Tato kapitola se zabývá konkrétní implementací v jazyce C# založené na návrhu z kapitoly čtvrté. Začíná celkovým pohledem na architekturu, rozdělení na jednotlivé moduly a organizaci tříd. Probírá zásadní datové struktury pro reprezentaci geometrie (bod, linie a polygon) a atributů (datový sloupec a řádek). Podrobněji představuje několik vybraných algoritmů pro určování vzdálenosti objektů nebo orientace vrcholů polygonu. Kapitulu uzavírá základní přehled metrik kódu a výsledných binárních souborů.

### 5.1 Architektura

Struktura aplikace vychází z objektového paradigmatu, které je v jazyce C# přímo vynuceno. Globální proměnné ani samostatně stojící funkce zde nejsou podporovány, veškerý kód musí být napsán uvnitř tříd. To sice přináší některé komplikace, například delší zápisy výrazů s proměnnými výčtových typů, na které se musí odkazovat včetně jména třídy, na druhé straně to programátor nutí dodržovat princip zapouzdření a psát kompaktnější kód, což uvedené nepohodlí bohatě kompenzuje.

Základním principem návrhu bylo oddělit aplikační logiku od uživatelského rozhraní. Celá aplikace představuje v terminologii Visual Studia jedno *Solution* (řešení). V rámci něj jsou definovány projekty s názvy *core* a *gui\_pda*. První je typu *Class library* (knihovna tříd) a seskupuje třídy realizující zmíněnou aplikační logiku. Výsledkem překladač tohoto projektu je DLL knihovna *core.dll*. Cílový framework knihovny je nastaven na Compact NET 3.5, takže je zaručeno, že veškerý kód bude spustitelný jak na mobilních zařízeních podporujících tuto verzi (kam WM 5 patří), tak na zařízeních podporujících plnohodnotný framework stejné verze (kam typicky patří počítače architektury x86). Druhý projekt je typu *Device application* (aplikace pro mobilní zařízení) a představuje okenní aplikaci spustitelnou na Windows Mobile. Obsahuje definici formulářů, komponent a pomocných metod pro jejich obsluhu.

V počátku vývoje se ukázalo, že i když je použití emulátoru PDA podstatně rychlejší než testování programu přímo ve fyzickém zařízení, stále je jeho použití pomalejší než práce s okenní aplikací určenou pro desktop. Proto bylo během vývoje a ladění DLL knihovny vytvořeno dočasné rozhraní určené výhradně pro desktop. To tvoří třetí projekt *gui\_desktop* typu *Windows application*. Toto rozhraní však není hlavním výstupem diplomové práce, nepodporuje proto kompletní množinu funkcí z kapitoly návrhu, je to jen jakýsi vývojový mezičlánek.

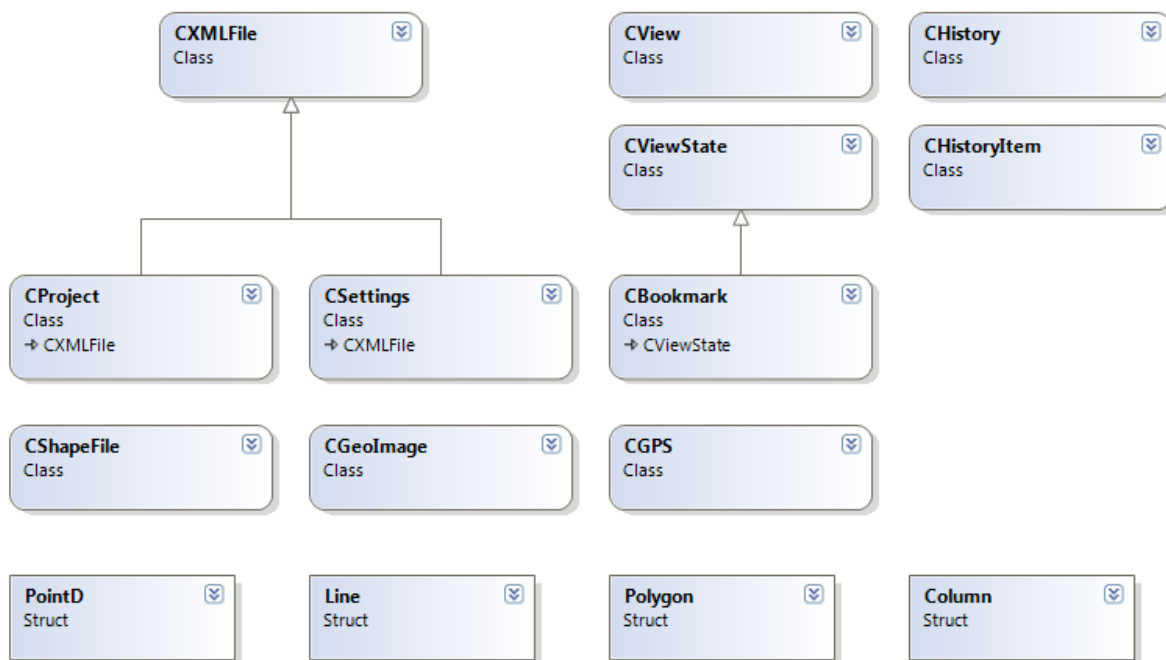
Použitý systém pojmenovávání identifikátorů a způsobu odsazování zdrojového kódu vychází z doporučení Microsoftu uvedených na jejich vývojářském portálu [11]. Jména interních proměnných v rámci metod, jejich parametrů a privátních vlastností tříd jsou tvořena dle takzvané *Camel case* notace, tedy první písmeno malé, v případě více slov každé další počáteční velké – například



`privateClassMember`. Privátní konstanty jsou psány kompletně velkými písmeny, s rozdělením případných víceslovných identifikátorů podtržítky (například `LINE_WIDTH`). Veškeré další identifikátory pak podle *Pascal case* – podobně jako *camel case*, ale i první písmeno je velké – například `void PublicMethod(int numericValue)`. Před názvy tříd je navíc přidáno velké písmeno C – `CShapeFile`. Názvy identifikátorů jsou psány v angličtině z důvodů usnadnění práce případným zahraničním pokračovatelům ve vývoji (v návaznosti na zveřejnění zdrojových kódů v anglických GIS katalogích). U podmínkové konstrukce `if - else` a v cyklech `for` i `while` je jejich tělo vždy uzavřeno ve složených závorkách, i když jde pouze o jediný řádek. Zjednodušuje to práci u eventuálního přidávání příkazů a eliminuje chyby z přehlednutí v případě více vnořených bloků. Každý blok složených závorek začíná na samostatném řádku. Vyrůstá tím sice délka zdrojového kódu, což je vyváženo jeho větší přehledností, která opět vede ke snížení pravděpodobnosti chyby z přehlédnutí.

### 5.1.1 Přehled tříd

Jádro aplikace je tvořeno jedenácti třídami a čtyřmi prostými strukturami viz zjednodušený Obrázek 14 vygenerovaný Visual Studiem. Kompletní diagram včetně seznamu metod a vlastností je v příloze 1. Popis významů jednotlivých tříd následuje dále.



Obrázek 14: zjednodušený přehledový diagram tříd a struktur.

Třída `CXMLFile` zapouzdřuje základní operace pro práci s XML soubory jako jejich načítání z disku a získání hodnot a atributů z nich, ať už jednotlivě nebo jako kolekci (například seznam

vrstev). Potomky jsou třídy `CProject` a `CSettings`. První z nich sdružuje data o zpracovávaném projektu, druhá uchovává nastavení samotné aplikace. Konfigurační soubor `settings.cfg` může vypadat následovně:

```
<?xml version="1.0" encoding="utf-8"?>
<settings>
  <hlcolor>255,0,128</hlcolor>
  <bgcolor>255,255,255</bgcolor>
  <hlwidth>3</hlwidth>
  <snapdist>5</snapdist>
  <port>COM1</port>
  <baudrate>9600</baudrate>
  <fillpoly>True</fillpoly>
</settings>
```

Tag `hlcolor` určuje barvu zvýraznění linie vybraného segmentu definováním jejich složek v pořadí červená, zelená, modrá. Stejná sémantika je použita i pro značku `bgcolor` určující barvu pozadí mapového výřezu. `Hlwidth` udává tloušťku vybraného liniového segmentu v pixelech, `snapdist` pak toleranci při výběru objektů na displeji, rovněž v pixelech. Tolerance má za úkol zjednodušit provádění výběrů, protože s dotykovým perem je obtížné trefit se na malé objekty na pixel přesně. Konkrétní algoritmy použité pro výběry jsou popsány dále v podkapitole 5.3. Značky `port` a `baudrate` se vztahují k nastavení GPS přijímače, jde o číslo sériového portu a přenosovou rychlost. Poslední položka, `fillpoly`, rozhoduje, zda se budou polygony vykreslovat i s výplní, nebo jen jako obrys.

Jednotlivé vektorové vrstvy jsou reprezentovány třídou `CShapeFile`, grafický podklad pak `CGeoImage`. `CShapeFile` obsahuje metody pro načítání a ukládání na disk, provádění výběrů a operací s objekty, jejich změnu, přidávání, mazání a vykreslování. Sdružuje geometrii i atributy. O jejich reprezentaci pojednává podkapitola 5.2. `CGeoImage` uchovává rastrová data obrazu společně s georeferenčními údaji. Stará se o načítání z disku (jak grafiky, tak georeference) a vykreslování.

Historii operací implementuje třída `CHistory`, představující rozšířený zásobník tříd `CHistoryItem`, které odpovídají uživatelem provedeným operacím. Uchovává se identifikace dotčeného objektu a podle typu operace detaily potřebné pro její navrácení.

Údaje vztahující se k aktuálnímu mapovému pohledu zapouzdřuje třída `CView`. Patří sem především použité měřítko, pozice okna nad mapou a minimální a maximální souřadnice určující celkovou plochu projektu. Z metod jsou to pak operace s měřítkem (oddálení a různé metody přiblížení) a obousměrné přepočty souřadnic mezi prostorem displeje a projektu. S využitím třídy `CViewState` ukládá jednotlivé pohledy na zásobník, aby se k nim mohl uživatel později vracet.

Rozšířením `CViewState` vzniká `CBookmark` sloužící k definici pojmenované pohledové záložky. Pohled je definován měřítkem a pozicí mapového okna, záložka navíc přidává jméno.

CGPS sdružuje operace pro komunikaci s GPS přijímačem. Kromě základních metod pro start a zastavení komunikace generuje při příjmu a úspěšném dekódování věty událost oznamující změnu interního stavu. Na ni je navázána funkce uživatelského rozhraní pro překreslování GPS kurzoru. Přes veřejné proměnné `Latitude`, `Longitude` a `Altitude`, respektive `X` a `Y` je přístupná aktuální pozice přijímače. První tři značí zeměpisnou šířku, délku a nadmořskou výšku v systému WGS-84, další dvě pak pozici podle S-JTSK. Přepočít je realizován podle algoritmu uvedeného v učebním textu [12]. Pro případ ztráty signálu nebo úmyslnou ruční korekci je v aplikaci zaveden takzvaný manuální režim – po jeho aktivaci se poloha GPS kurzoru neurčuje podle dat z přijímače, ale přímo dotykovým perem na obrazovce. Aby nedošlo ke špatné interpretaci dat uživatelem, odlišuje se kurzor v manuálním režimu jinou barvou kurzoru a značkou ve stavovém řádku. Uvnitř třídy je mód odlišen veřejným příznakem `Manual` typu `bool`.

## 5.2 Datové struktury

Datově nejnáročnější strukturou v programu je kolekce tříd `CShapeFile` představující seznam vektorových vrstev. Třída v sobě sdružuje jak geometrii, tak atributy objektů. Původní záměr byl použít pro manipulaci s `.shp` a `.dbf` soubory některou z hotových, volně šířených knihoven dostupných na internetu. Po prozkoumání několika z nich se ukázalo, že pro potřeby tohoto projektu se jedná o příliš rozsáhlé a robustní řešení. Metody manipulující s daty jsou tedy původní, vytvořené podle online dokumentací [9] a [10].

### 5.2.1 Geometrie

Základní datovou jednotku, která se využívá ve všech typech prostorových entit, tvoří bod se dvěma souřadnicemi určenými desetinnými čísly s dvojitou přesností (datový typ `double` uložený na 64 bitech). Jedná se o strukturu `PointD` ze zjednodušeného diagramu tříd. Název byl zvolen podle struktur vestavěných v CNF – `Point` pro celočíselné integer souřadnice a `PointF` pro desetinné typu `float`.

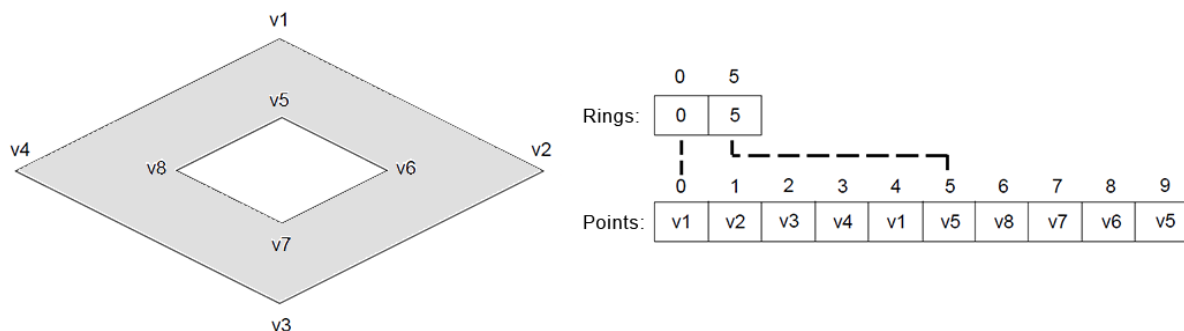
```
struct PointD
{
    double X; // X souřadnice
    double Y; // Y souřadnice
}
```

Dvourozměrná struktura linie se v terminologii anglické technické dokumentace nazývá *multiline*, takže česky bychom spíše měli hovořit o multi-linii nebo multi-čáře, budeme se ale držet

kratšího, jednoslovného termínu. Důvodem pro složitější anglické označení je skutečnost, že jeden logický liniový objekt může sestávat z více oddělených částí, které na sebe prostorově nenavazují. Příkladem může být mapová značka mostu – dvě rovnoběžné úsečky, která ale tvoří jediný celek. Tyto jednotlivé části pak sestávají z řady bodů a mohou se navzájem křížit. Protože jsou body všech úseků uloženy v jednom poli bez oddělovače, je součástí struktury linie údaj o počtu částí (celočíslný typ `integer` uložený na 32 bitech) a pole indexů jejich začátků. Posledním údajem je čtyřprvkové pole `box[]` určující prostorový ohraničující obdélník (takzvaný *bounding box*) linie se sémantikou `box[0]` = nejmenší X souřadnice, `box[1]` = nejmenší Y souřadnice, `box[2]` = největší X souřadnice a nakonec `box[3]` = největší Y souřadnice. V programu je linie reprezentována strukturou `Line`:

```
struct Line
{
    double[4] box;      // ohraničující oblast
    integer numParts;    // počet částí
    integer numPoints;   // celkový počet bodů
    integer[numParts] parts; // indexy počátků částí
    List <PointD> points; // všechny body linie
}
```

Nejsložitější použitou geometrickou strukturou je polygon. Ten může, podobně jako linie, sestávat z více částí, které se nazývají okruhy. Situaci ale komplikuje přípustné vnořování okruhů, což je způsob reprezentace „ploch s dírami“, například plán zemědělských oblastí s vyňatými chráněnými územími. V tomto případě mluvíme o vnějších a vnitřních okruzích, kdy vnitřní je vnořený ve vnějším. Kromě této rozšiřující vlastnosti může, stejně jako u linií, jeden logický polygon tvořit více fyzicky oddělených částí, příkladem budiž politická mapa souostroví tvořícího jeden stát.



Obrázek 15: ukázka uložení polygonu s jedním vnějším a jedním vnitřním okruhem.

Okruh je definován jako uzavřená posloupnost bodů, kde poslední se musí přesně shodovat s prvním a nesmí tvořit smyčky ani křížení. Zda jde o vnitřní nebo vnější okruh se rozpoznává z pořadí uložení bodů. Jsou-li body poskládány po směru hodinových ručiček, jedná se o vnější

okruh, v opačném případě o vnitřní. K detekci této vlastnosti slouží algoritmus z podkapitoly 5.3.3. Všechny body polygonu jsou opět uloženy v jednom poli s odděleným seznamem indexů počátků jednotlivých okruhů. Opakování počátečních bodů na koncích popisů úseků je pro interní potřeby programu nežádoucí, proto jsou tyto body při načítání z disku do paměti přeskakovány, naopak při zápisu zase přidány, aby data zůstala konzistentní a korektní. Situaci s jedním vnějším a jedním vnitřním okruhem znázorňuje Obrázek 15, zápis struktury Polygon vypadá takto:

```
struct Polygon
{
    double[4] box;        // ohraničující oblast
    integer numRings;     // počet okruhů
    integer numPoints;    // celkový počet bodů
    integer[numRings] aspects; // orientace okruhů
    integer[numRings] rings; // indexy počátků okruhů
    List <PointD> points; // všechny body polygonu
}
```

## 5.2.2 Atributy

Pro uchovávání atributů jsou použity dva typy struktur – charakteristika sloupce Column a řádek dat jako pole řetězců, s tím, že počet jeho položek odpovídá počtu sloupců. Přístup, kdy jsou všechny hodnoty uloženy jako textové řetězce (tedy i čísla a logické hodnoty) sice není sémanticky úplně v pořádku, ale v souboru na disku je stanoven stejný systém, navíc v aplikaci se hodnoty využívají pouze k zobrazování, nikoliv jako součást výrazů nebo dalších výpočtů.

```
struct Column
{
    string Name;
    char Type;
    byte Length;
    byte Decimals;
}
```

Name značí název sloupce a Type jeho typ. Podle specifikace je to jedno z písmen N, C, D nebo L s významem číslo (jak desetinné, tak celé), řetězec znaků, datum a logická hodnota. Length určuje celkovou maximální délku řetězce s hodnotou, Decimals pak počet desetinných míst u desetinných čísel. Při uživatelských změnách hodnot atributů je důležitá důkladná kontrola, zda vstup spadá do povoleného rozsahu hodnot sloupce. Ověřování je navázáno na událost Validating vstupního pole, která je systémem generována automaticky v případě, že textový kurzor opustí pole, typicky při kliknutí na jiný prvek uživatelského rozhraní.

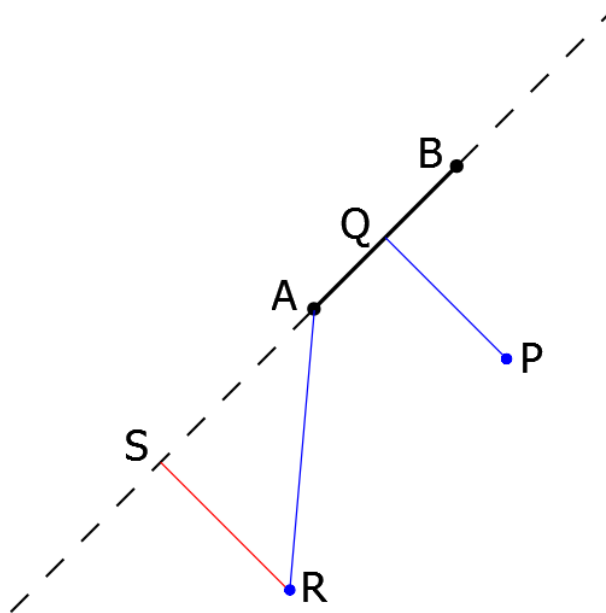
## 5.3 Vybrané algoritmy

Z hlediska návrhu tříd nepřináší program nic nezvyklého, oddělení aplikační logiky a uživatelského rozhraní je obecně osvědčená praktika. Datové struktury kopírují formát diskových souborů, potažmo dokumentaci k nim. Dále proto následuje bližší pohled na několik konkrétních algoritmů.

První dva se týkají problému nalezení nejbližšího objektu po kliknutí na displej, a sice úsečky a polygonu. Spadá sem i hledání nejbližšího bodu, což je však záležitost triviální – jednoduše se počítá Euklidovská vzdálenost mezi body vrstvy a bodem dotyku na displeji. Třetí ukázka se vztahuje k rozlišování vnějších a vnitřních okruhů při vykreslování polygonů – podle toho, zda jsou body orientovány po nebo proti směru hodinových ručiček.

### 5.3.1 Výpočet vzdálenosti bodu od úsečky

Typickou úlohou analytické geometrie je výpočet vzdálenosti bodu od přímky. Pro naše účely je však potřeba postup modifikovat, protože hledáme vzdálenost od jednotlivých liniových úseků, tedy úseček. Algoritmus pro přímku je nenásledující: ze zkoumaného bodu se vytvoří kolmice na přímku a určí se vzdálenost zkoumaného bodu a nového průsečíku. Modifikace pro úsečku spočívá v detekci polohy průsečíku. Leží-li mezi koncovými body úsečky, je vše v pořádku a postupuje jako pro přímku. Pokud ale průsečík leží na prodloužení úsečky na pomyslnou přímku, vně koncových bodů úsečky, počítáme vzdálenost nikoliv k průsečíku, ale k bližšímu koncovému bodu úsečky. Situaci znázorňuje Obrázek 16. Zkoumáme vzdálenost bodů  $P$  a  $R$  od úsečky  $AB$ . Pro bod  $P$  postupujeme podle algoritmu pro přímku, čímž dostaneme průsečík  $Q$  a výsledkem je vzdálenost  $|PQ|$ . Pro bod  $R$  bychom ale dostali zdánlivý průsečík  $S$  a vzdálenost  $|RS|$ , což není korektní výsledek, protože nás zajímá vzdálenost  $|AR|$ .

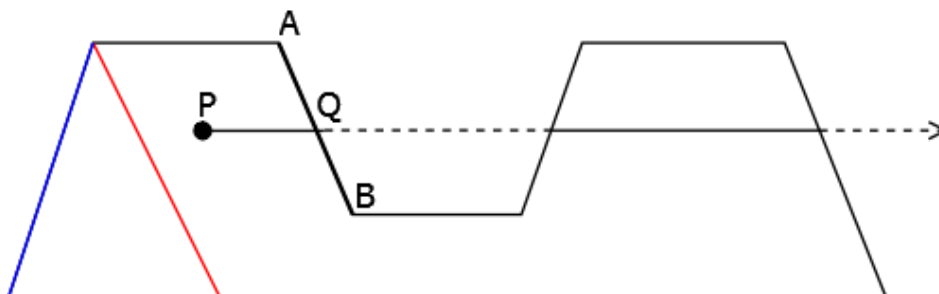


Obrázek 16: princip zkoumání vzdálenosti bodu od úsečky.

Postup inspirovaný [13] je následující: známe tedy polohu bodů  $A$ ,  $B$  a  $P$ . Polohu  $Q$  vypočteme z parametrické rovnice přímky  $Q = A + u(B - A)$ , kde  $u$  je reálný parametr a  $B - A$  směrový vektor přímky. Protože úsečka  $PQ$  je na  $AB$  kolmá, je skalární součin vektorů  $A - B$  a  $P - Q$  roven 0. Parametr  $u$  určuje, kolikrát násobek směrového vektoru musíme k bodu  $A$  přičíst, abychom dostali bod  $Q$ . Pokud je jeho velikost záporná, znamená to, že  $Q$  by na přímce leželo pod  $A$ . Hodnota větší než 1 naopak značí, že  $Q$  by bylo na přímce nad  $B$ . Hodnoty parametru proto ořežeme na uzavřený interval  $\langle 0; 1 \rangle$ , což nám zaručí, že pomyslné průsečíky ležící vně úsečky se přesunou do jejich krajních bodů. Se znalostí parametru a rovnic přímek vypočítáme souřadnice průsečíku a následně vzdálenost od zkoumaného bodu  $P$ .

### 5.3.2 Test polohy bodu uvnitř polygonu

Pro ověřování, zda bod leží uvnitř polygonu, se v aplikaci používá varianta založená na takzvaném paprskovém algoritmu publikovaném v knize [14]. Představme si pozorovatele, který stojí uvnitř polygonu. Vydá se směrem doprava a překročí hranici polygonu. Tím se ocitne vně. Jde dál a překročí hranici znova, takže je zase uvnitř. Rozhodující je tedy počet překročení hranice – při lichém jsme začínali uvnitř, při sudém vně polygonu. V praxi ale není potřeba počítat jednotlivé přechody, stačí pokaždé negovat hodnotu logické proměnné značící „uvnitř“ s nepravdivou počáteční hodnotou. Situaci ukazuje Obrázek 17.



Obrázek 17: princip algoritmu pro test polohy bodu uvnitř polygonu.

Algoritmus postupně bere dvojice po sobě následujících bodů, které tvoří úseky hranice polygonu, a testuje, zda  $Y$  souřadnice startovního bodu leží mezi  $Y$  souřadnicemi těchto dvou bodů. Pokud ano, jedná se o kandidátní situaci na „překročení hranice“. Je ale ještě potřeba ověřit, zda hrana opravdu leží v dráze paprsku. Modrá a červená hrana v obrázku požadavky na  $Y$  souřadnice splňují, přesto netvoří hranu, kterou by paprsek prořel. Kontrola spočívá v dosazení  $Y$  souřadnice bodu  $P$  do směrnice rovnice přímky tvořené koncovými body  $A$ ,  $B$  hrany polygonu. Výsledná souřadnice  $X'$  imaginárního bodu  $Q$  musí být větší než  $X$  souřadnice bodu  $P$ , tedy že  $P$  leží nalevo od  $Q$ .

### 5.3.3 Určení orientace posloupnosti bodů

Tento algoritmus se využívá u polygonů složených z více částí – orientace určuje, zda jde o okruh vnitřní, nebo vnější.

Během testu se počítá obsah polygonu a podle znaménka výsledku se odvodí orientace bodů, idea pochází z příspěvku [15]. Záporný obsah značí body uspořádané po směru hodinových ručiček, nezáporný naopak. Postupně jsou procházeny dvojice po sobě následujících bodů (tedy hrany polygonu) a k celkovému obsahu je přičítána plocha lichoběžníku tvořeného těmito body a jejich průměty do osy x. V obvyklém geometrickém vzorci pro obsah lichoběžníku  $S = (základna1 + základna2) * výška / 2$  se základny změny na Y souřadnice bodů a výška na rozdíl jejich X souřadnic. Leží-li druhý bod hrany nalevo od prvního, vyjde obsah oblasti pod hranou záporný, takže se od celkové plochy odečte (pro druhý bod napravo od prvního vyjde kladný přírůstek). Uvedené platí pro body orientované po směru hodinových ručiček, u opačné posloupnosti se znaménka prohodí, což ve výsledku znamená i zápornou hodnotu pro celkovou sumu.

## 5.4 Metriky kódu

Na závěr kapitoly je uveden přehled základních metrik kódu a výsledných binárních souborů, viz následující Tabulka 1.

	<b>core.dll</b>	<b>gui_pda.exe</b>	<b>gui_desktop.exe</b>
<b>Velikost po překladu:</b>	45 kB	70 kB	85 kB
<b>Řádky kódu:</b>	2903	2675	2883
<b>Řádky komentářů:</b>	408	695	592
<b>Řádky celkem:</b>	3311	3370	3475

Tabulka 1: přehled základních metrik kódu a binárních souborů.

Překlad probíhal ve Visual Studiu 2008 Professional na PC s dvoujádrovým procesorem Intel Core2Duo a operačním systémem Windows 7 Professional. Operační systém pro PDA byl Windows Mobile 5. Velikost binárních souborů je potřeba hodnotit v kontextu velikosti NET Frameworku, který je pro chod nezbytný a na disku zabírá o několik řádů více místa. Na stránce pro stažení uvádí Microsoft pro desktopovou verzi náročnost „až 500 MB volného místa“, pro mobilní je známa jen velikost instalačního balíčku 33 MB.



## 6 Praktické testování

Nedílnou součástí vývoje prakticky zaměřeného programu je i jeho ověření v reálných podmínkách. Účelem je otestovat, zda navržené uživatelské rozhraní programu umožňuje efektivní ovládání všech funkcí a zda je výkon a odezva aplikace dostatečná.

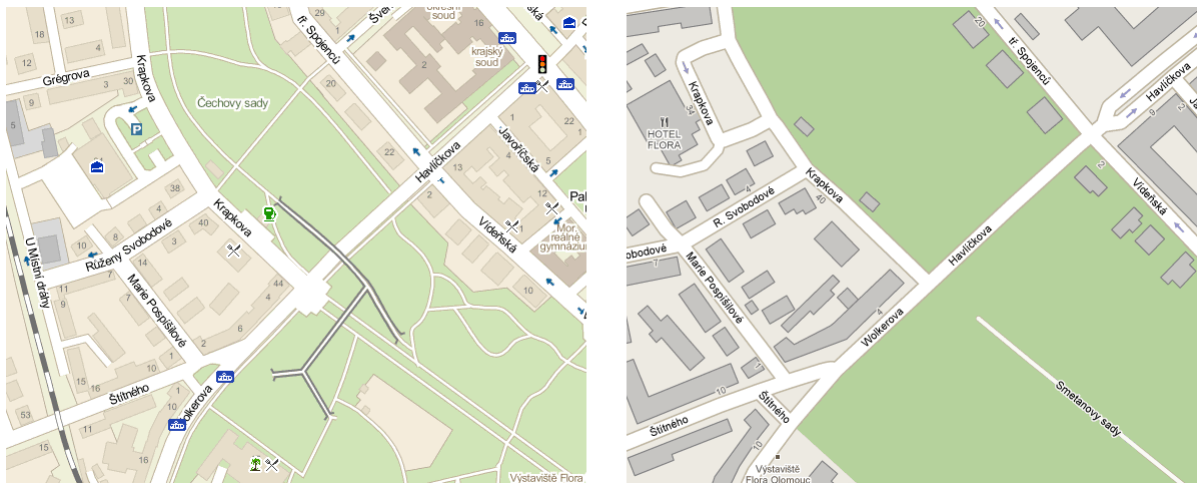
Jako obsah praktického testu bylo určeno zmapování mostu spojujícího dva největší olomoucké parky – Smetanovy a Čechovy sady. Jedná se o komplikovanou mostní konstrukci s několikanásobným větvením a možnostmi nástupu po rampách nebo schodech (Obrázek 18). V roce 2006 byl tento most rekonstruován a jeho délka zvětšena. Absolutní geografický rozsah objektu není sice příliš velký, ale dokumentace jeho stavu po rekonstrukci není veřejně dostupná. Mapový server <http://mapy.cz> zachycuje stav před rekonstrukcí, na <http://maps.google.com> není most zobrazen vůbec, viz Obrázek 19.



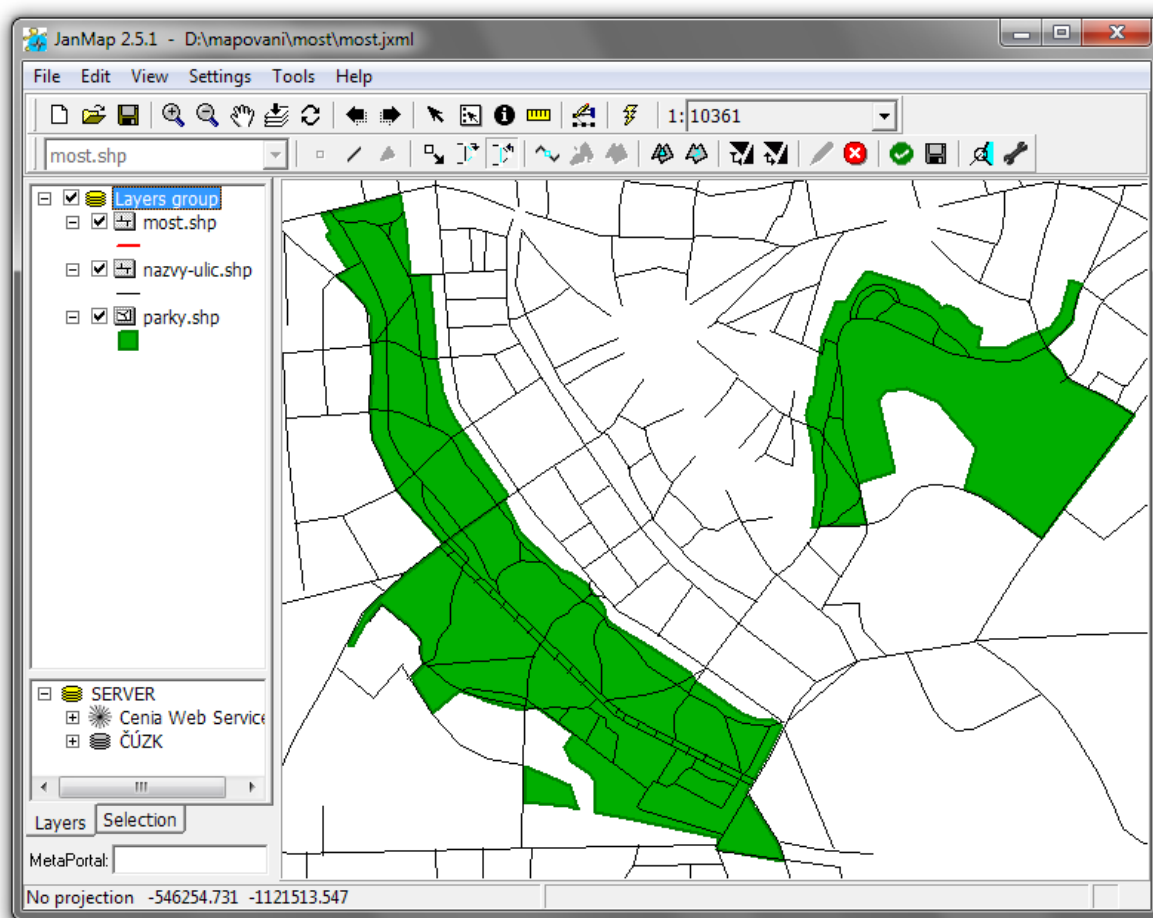
Obrázek 18: snímek mostu, nástup přes schody.

Úkolem tedy bylo zaměřit koncové a lomové body mostu společně s úhly sklonů ramp a schodů. K orientačnímu měření sklonů byla použita aplikace *iHandy Level* pro mobilní telefon Apple iPhone, který je vybaven citlivými snímači polohy. Příprava na měření spočívala ve vytvoření nového projektu *most* v českém, volně dostupném, desktopovém programu Janitor. Projekt obsahoval dvě hotové vrstvy *nazvy-ulic* a *parky* pocházející ze studentského projektu vypracovaného na Katedře geoinformatiky Univerzity Palackého v Olomouci. První (liniová) vrstva představovala uliční síť a druhá (polygonová) zobrazovala plochy městských parků. Třetí, nově vytvořenou vrstvou byl *most*,

tedy liniová vrstva s jedním číselným atributem sklon. Rozhraní programu a otevřený projekt ukazuje Obrázek 20. S pomocí interní čtečky v notebooku bylo všech devět souborů (tři vrstvy po třech souborech – .shp, .shx a .dbf) nahráno na SD kartu, která byla vložena do PDA.



Obrázek 19: zobrazení mostu na serverech mapy.cz a maps.google.com, stav k 15. 5. 2010.



Obrázek 20: náhled programu Janitor s otevřeným projektem.

Další projekt byl vytvořen přímo v zařízení. Kromě tří vrstev zkopírovaných z PC byla přidána i jedna pohledová záložka *park* zobrazující území, na kterém se most zhruba rozkládá. Obsah projektového souboru byl následující:

```
<?xml version="1.0" encoding="utf-8"?>
<project>
  <scale>1.901703</scale>
  <offsetx>1614</offsetx>
  <offsety>1498</offsety>
  <coordsys>SJTSK</coordsys>
  <layers>
    <layer visible="True" active="True" path="\Storage
      Card\most" color="255,0,0" />
    <layer visible="True" active="False" path="\Storage
      Card\nazvy-ulic" color="0,0,0" />
    <layer visible="True" active="False" path="\Storage
      Card\parky" color="0,130,0" />
  </layers>
  <bookmarks>
    <bookmark offsetx="770" offsety="688" scale="3.897032"
      name="park" />
  </bookmarks>
</project>
```

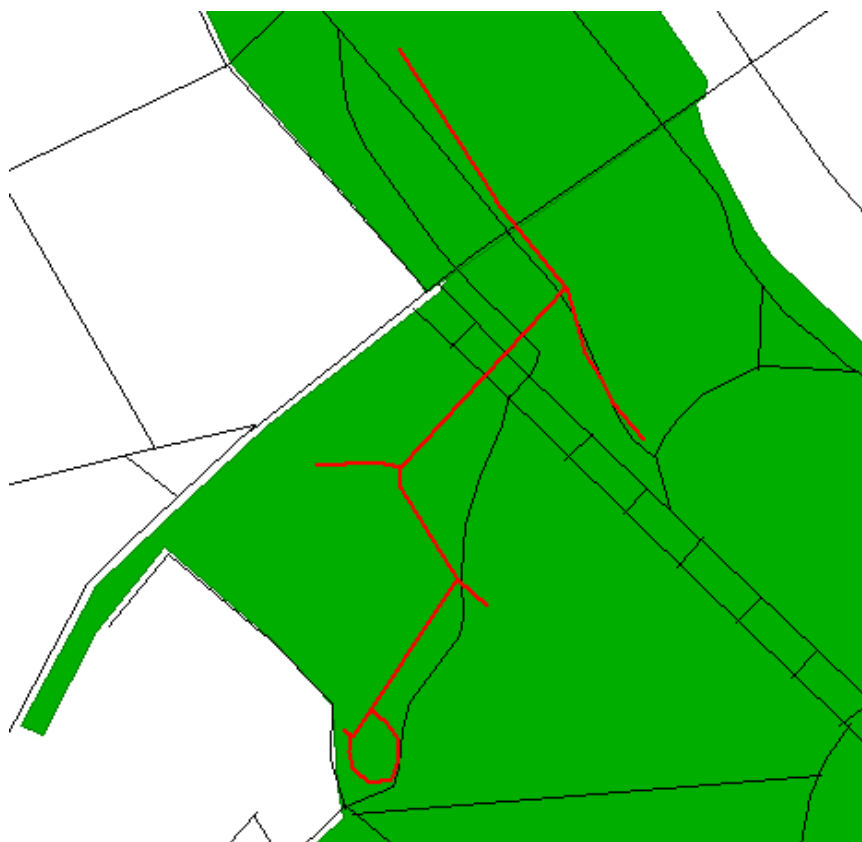
Vlastní měření proběhlo 19. 5. 2010 v čase 13.11 až 13.58. Teplota vzduchu byla 9,6 °C, obloha zatažená s drobným, vytrvalým mrholením. Podle webu [chmi.cz](http://chmi.cz) byla vlhkost vzduchu v nejbližší meteorologické stanici v Přerově 92%. Jednalo se tedy nejen o test aplikace, ale i o odolnost samotného zařízení. K manipulaci s PDA při takových podmínkách byly získány dva poznatky: pro ochranu přístroje by bylo vhodné použít plastový ochranný obal a současně ho vybavit šňůrou pro zavěšení na krk. Kapky vody na displeji přístroji jistě neprospívají a navíc snižují čitelnost displeje. Čitelnost, možná i citlivost displeje na dotyk by mohl obal sice také snížit, nicméně před nepřízní počasí by přístroj ochránil určitě. Výsledný přínos by tedy bylo nutné ověřit s konkrétním obalem. Zavěšení na krk je praktické v případě střídavé manipulace s jiným zařízením, v tomto případě měřidlem sklonu. Neustálé vkládání a vyndávání z kapsy bundy nebylo pohodlné a navíc hrozilo, že přístroj může vypadnout na zem a rozbít se.

Z hlediska práce s aplikací byl odhalen jeden problém: pokud je vytvářen nový objekt a uživatel před jeho uložením zvolí některou z funkcí pro práci s mapovým výřezem, objekt se resetuje, tedy nenávratně smaže. Toto působí komplikace především v případě, kdy se uživatel dostane fyzicky mimo oblast právě zobrazovanou na displeji a chce pohled buď posunout, nebo oddálit.

Co se týče příjmu signálu z GPS, byl po počáteční lokalizaci družic bezproblémový i přes částečné zastínění mostu vzrostlými stromy. Při mapování trvajícím necelou hodinu se ale baterie přístroje vybila z 60%. U referenčního PDA je to nejspíš způsobeno stářím a nejasnou historií baterie,

protože přístroj pochází z bazaru. Nová baterie by pravděpodobně vydržela déle, nicméně konkrétní hodnoty (se zapnutou GPS a stále rozsvíceným displejem) by bylo potřeba ověřit před cestou do terénu. Pro déletrvající měření by mohlo být praktické vlastnit přídavný externí zdroj či více interních baterií pro výměnu. Další zajímavostí bylo, že interní GPS mobil v PDA zvládal komunikaci pouze rychlostí 4.800 Bd, tedy přesně podle specifikace NMEA protokolu. Externí modul připojený přes Bluetooth pracoval i při 57.600 Bd.

Po ukončení mapování byly změněné soubory `most.*` z paměťové karty nahrány zpět do PC. Náhled výsledku v prostředí Janitoru ukazuje Obrázek 21. Naměřené sklony byly 27,5 stupně pro schody a 4,8 až 5,6 stupňů pro rampy.



Obrázek 21: výsledek mapování – červená čára představuje most.

Některé podněty tedy vzešly i z tohoto krátkého praktického testu, pro získání dalších by bylo potřeba provádět komplexnější operace – především pracovat s více vrstvami. To je také námět pro další testování za lepšího počasí a s novou baterií.

# Závěr

Cílem této práce bylo prozkoumat současný stav mezi aplikacemi pro kapesní počítače sloužícími pro podporu mapování v terénu. I přes dostupnost všech technických vymožeností, jako jsou globální systémy pro určování polohy nebo metody dálkového průzkumu Země, je totiž někdy potřeba vyrazit ven, do terénu, a data získat ručně přímo na místě.

Technická zpráva začíná objasněním a definováním prerekvizitních pojmů z oblasti geografických informačních systémů. Od obecné charakteristiky GIS a struktury zpracovávaných dat se přes způsoby a možnosti jejich získávání dostává k příležitostem využití a principům fungování globálních polohovacích systémů. Zabývá se i možnostmi zpracování získaných dat na desktopu. Nakonec se přes pojem digitální zápisník dostává ke kapesním počítačům. Všímá si platformy dostupných v tomto segmentu výpočetní techniky z hlediska jejich rozšíření na trhu a možností vývoje vlastního software pro ten který systém.

Z analýzy nejkompaktnějšího dostupného programu ArcPad pro cílovou platformu Windows Mobile vzešly podněty, které byly ve čtvrté kapitole doplněny a precizovány. Bylo potřeba zvolit podporované datové formáty, určit nezbytné funkce programu a rozvrhnout uživatelské rozhraní tak, aby bylo možné je v plné míře využívat.

Pátá kapitola přiblížila vnitřní strukturu aplikace, tedy principy návrhu tříd stejně jako vnější strukturu v podobě uživatelského rozhraní. Blíže je zde popsáno několik klíčových algoritmů vycházejících z postupů analytické geometrie v rovině.

Ověřením skutečné funkčnosti a použitelnosti se zabývala poslední část práce. Jednalo se o kompletní proces mapování reálného geobjektu, tedy příprava podkladů na stolním počítači, jejich načtení na PDA, měření v zájmové lokalitě a nakonec přenesení výsledků zpět do PC.

Právě poslední kapitola ukázala, že aplikaci lze použít pro řešení skutečného problému. Současně z ní vzešlo i několik postřehů jak k samotnému průběhu mapování s použitím mobilní výpočetní techniky v nepříznivých povětrnostních podmínkách, tak k samotnému programu. Úspěšné vyzkoušení aplikace však neznamená, že by její vývoj byl u konce. Nabízí se mnoho dalších možností, jak ji dále vylepšit. Jako první se nabízí rozšíření množiny podporovaných datových formátů jak pro načítání, tak pro ukládání dat. Podobně je možné doplnit další volitelné souřadné systémy. Implicitní formát shapefile a systémy WGS-84 a S-JTKS použité v současné verzi aplikace byly vybrány jako nejpoužívanější a proto teoreticky vyhovující největší podmnožině potenciálních uživatelů. Ti by mohli díky anglickému rozhraní programu přicházet nejen z České či Slovenské republiky. Stejně tak by mohlo zveřejnění aplikace (včetně zdrojových kódů) na světových oborových serverech přispět k jejímu dalšímu vývoji. Malá velikost, podpora standardních datových formátů a volná dostupnost kompletních zdrojových kódů patří mezi hlavní přínosy vytvořené aplikace.

# Literatura

- [1] Hrubý, M.: Geografické Informační Systémy (GIS). Studijní opora, FIT VUT, Brno, 2006. Dostupná online na <<http://perchta.fit.vutbr.cz:8000/vyuka-gis/uploads/1/GIS-final2.pdf>> (prosinec 2009).
- [2] Břehovský, M., Jedlička, K.: Úvod do geografických informačních systémů. Přednáškové texty, FAV ZČU, Plzeň. Dostupné online na <<http://gis.zcu.cz/studium/ugi/e-skripta/ugi.pdf>> (prosinec 2009).
- [3] Global Positioning System. Wikipedie, otevřená encyklopedie. Dostupná online na <<http://cs.wikipedia.org/wiki/GPS>> (duben 2010).
- [4] Martínek, J.: GPS a komunikační protokol NMEA. Série článků, 2006. Dostupné online na <<http://www.abclinuxu.cz/serialy/gps-a-komunikacni-protokol-nmea>> (prosinec 2009).
- [5] DePriest, D.: NMEA data. Webová stránka. Dostupná online na <<http://www.gpsinformation.org/dale/nmea.htm>> (prosinec 2009).
- [6] Neteler, M.: GIS GRASS - Praktická rukověť. Překlad Čepický, J. Trento, Itálie, 2003. Dostupná online na <[http://gama.fsv.cvut.cz/data/grasswikicz/grass\\_prirucka/grass\\_prirucka\\_0.4.pdf](http://gama.fsv.cvut.cz/data/grasswikicz/grass_prirucka/grass_prirucka_0.4.pdf)> (prosinec 2009).
- [7] Rapant, P.: Družicové polohové systémy. VŠB – TU, Ostrava, 2002. ISBN 80-248-0124-8.
- [8] Fried, J., Hansson, H. D., Linderman, M.: Getting Real – The smarter, faster, easier way to build a successful web application. 37signals, USA, 2009. ISBN 0578012812.
- [9] ESRI Shapefile Technical Description. White Paper, USA, 1998. Dostupný online na <<http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>> (prosinec 2009).
- [10] Bachmann, E.: Xbase Data file. Webová stránka. Dostupná online na <<http://www.clicketyclick.dk/databases/xbase/format/dbf.html>> (únor 2010).
- [11] .NET Framework General Reference – Naming Guidelines. Webová stránka. Dostupná online na <<http://msdn.microsoft.com/en-us/library/xzf533w0.aspx>> (leden 2010).
- [12] Hrdina, Z.: Transformace souřadnic ze systému WGS-84 do systému S-JTSK. Učební text, FEL ČVUT, Praha, 1997. Dostupný online na <[http://www.geospeleos.com/Mapovani/WGS84toSJTSK/WGS\\_JTSK.pdf](http://www.geospeleos.com/Mapovani/WGS84toSJTSK/WGS_JTSK.pdf)> (březen 2010).

- [13] Bourke, P.: Minimum Distance between a Point and a Line. Webová stránka, 1988.  
Dostupná online na <<http://local.wasp.uwa.edu.au/~pbourke/geometry/pointline>> (únor 2010).
- [14] Heckbert, P.: Graphics Gems IV. Academic Press, Boston, 1994.  
ISBN 0-12-336155-9.
- [15] Stephens, R.: Calculate a polygon's area in C#. Příspěvek na internetovém blogu. Dostupný online na <<http://blog.csharpHelper.com/2010/01/04/calculate-a-polygons-area-in-c.aspx>> (únor 2010).

# Seznam příloh

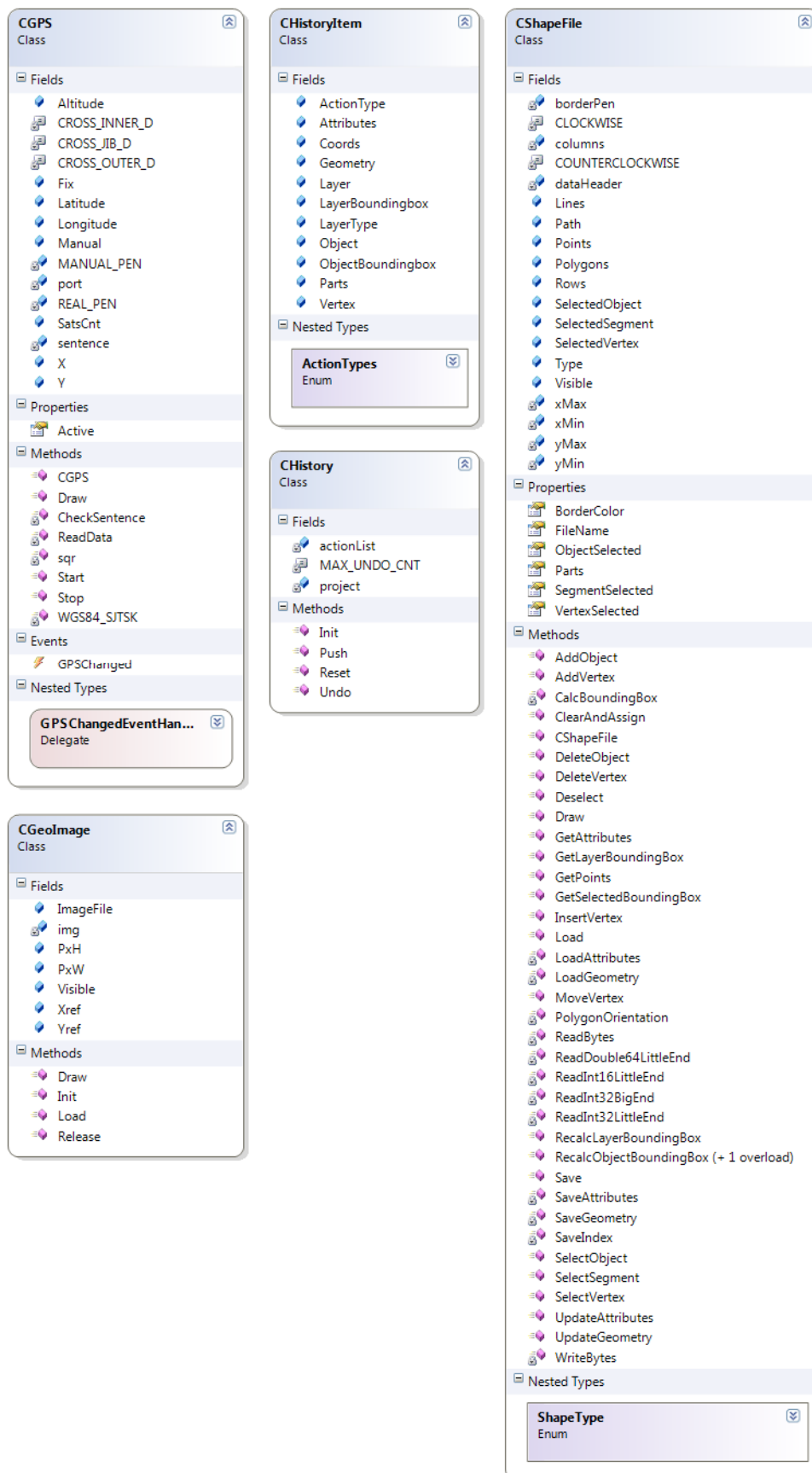
**Příloha 1:** Kompletní diagram tříd vygenerovaný Visual Studiem.

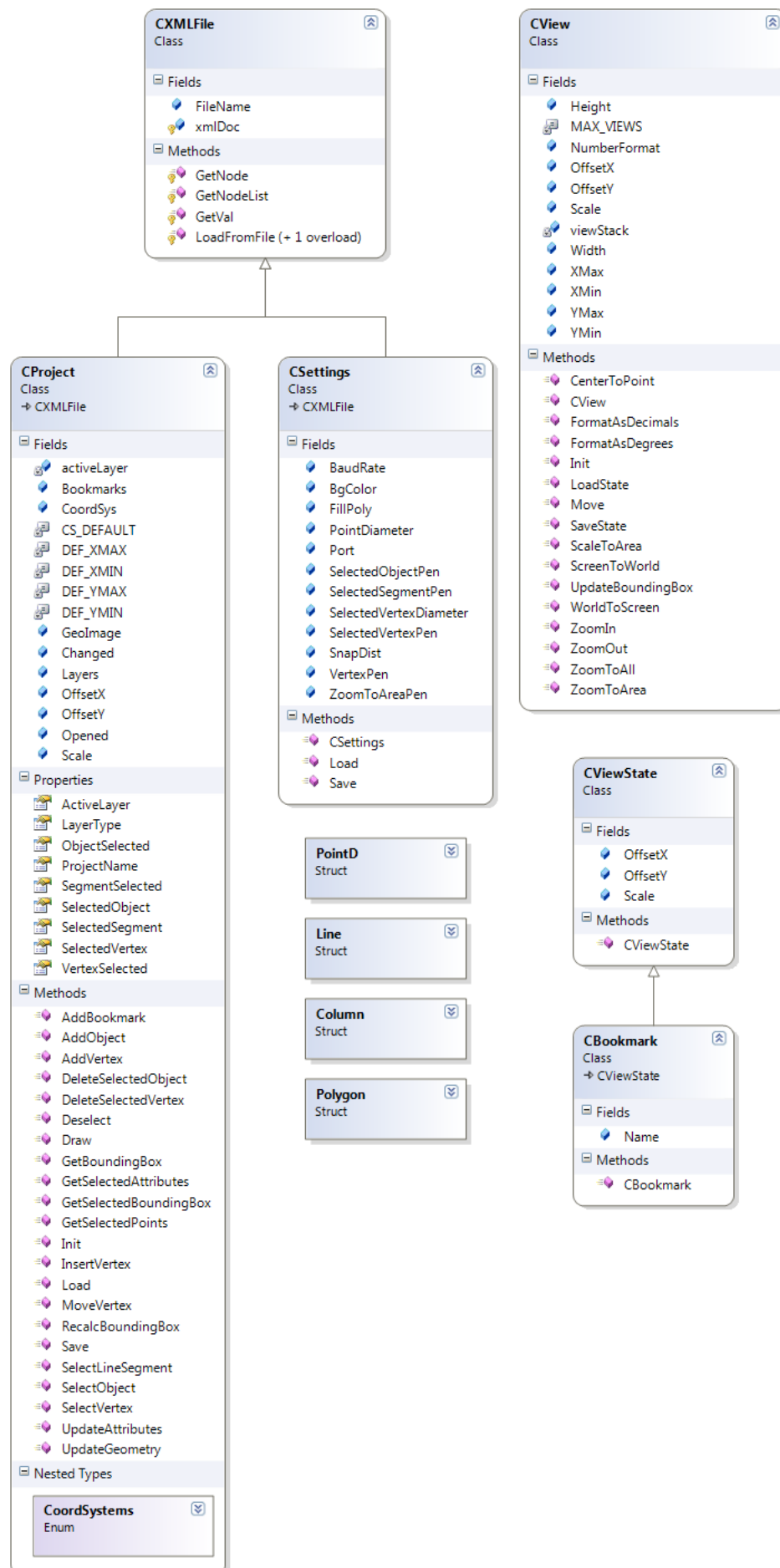
**Příloha 2:** Snímky obrazovek výsledné aplikace.

**Příloha 3:** CD se zdrojovými kódy, spustitelnou aplikací, ukázkovými daty a technickou zprávou v elektronické podobě.



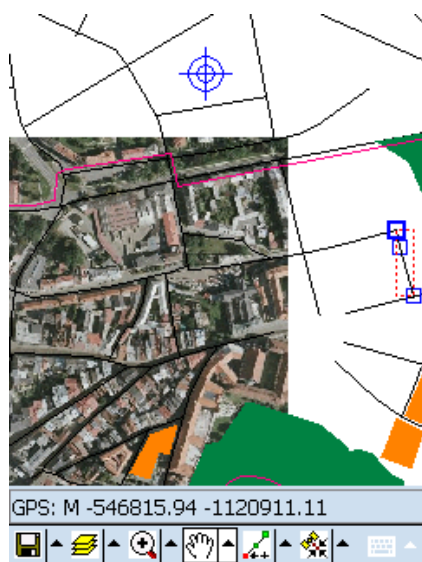
# Příloha 1: kompletní diagram tříd



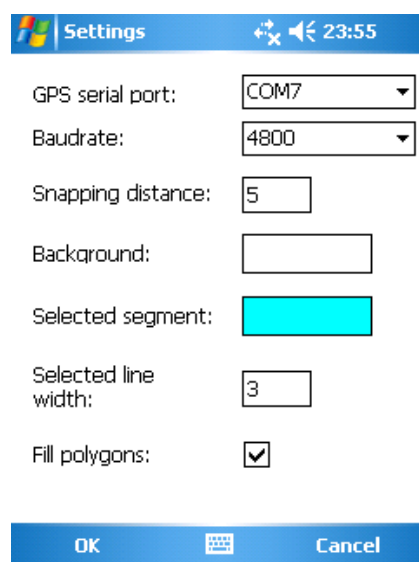


## Příloha 2: snímky obrazovek aplikace

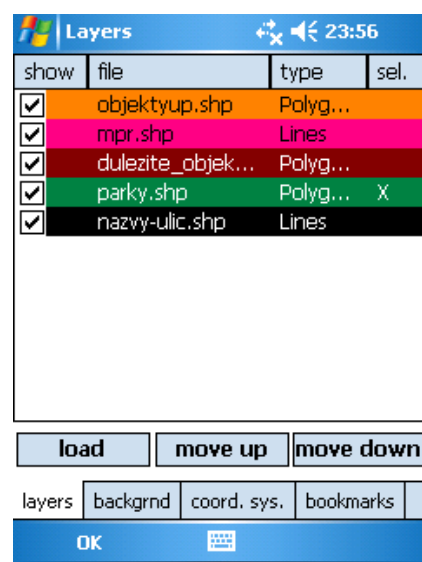
V tabulce následují snímky vybraných obrazovek z finální aplikace na PDA s rozlišením displeje 240 × 320 pixelů.



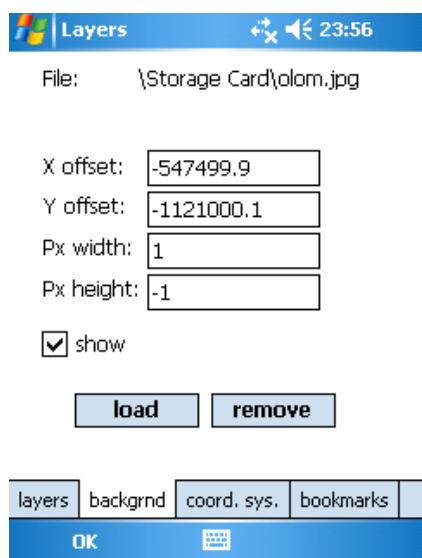
Hlavní mapové okno.



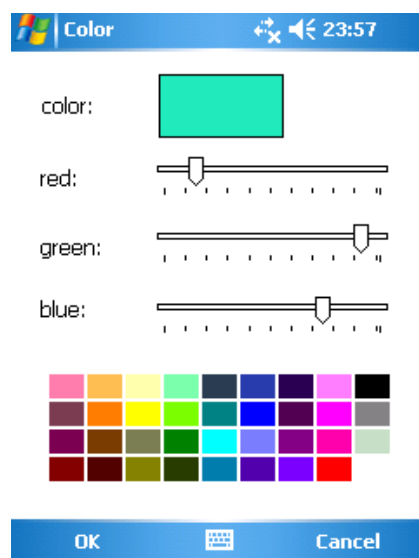
Nastavení aplikace.



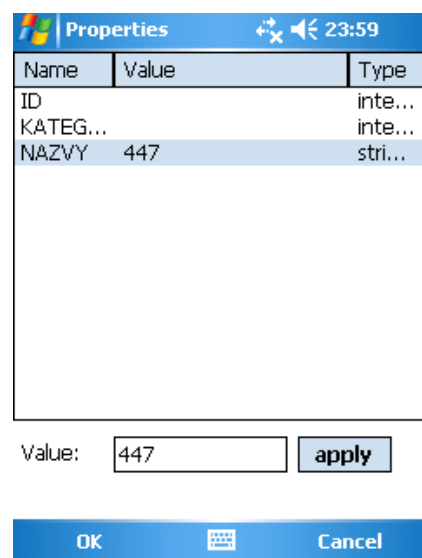
Seznam vrstev.



Parametry podkladu.



Výběr barvy



Atributy objektu.